THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Nicholas Joseph Cercone

TITLE OF THESIS: Representing Natural Language in Extended Semantic Networks

DEGREE FOR WHICH THIS THESIS WAS PRESENTED: Doctor of Philosophy

YEAR THIS DEGREE GRANTED: 1975

THE UNIVERSITY OF ALBERTA

REPRESENTING NATURAL LANGUAGE IN EXTENDED SEMANTIC NETWORKS

by

© Nick Cercone

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTING SCIENCE

EDMONTON, ALBERTA

FALL, 1975

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled "Representing Natural Language in Extended Semantic Networks", submitted by Nicholas Joseph Cercone in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

ABSTRACT

Computer use of natural language would be a fascinating phenomenon. Many obstacles must first be removed in order for this dream to become a reality. Not the least of these obstacles is a method for representing, in a formal and consistent way, the meaning content of language utterances. This representation of natural language utterances through meaning analysis is the central theme of this thesis.

The research reported herein is a development of some ideas concerning the representation of individual items of factual knowledge in a computer, where this knowledge is thought of as being conveyed in natural language. Many problems are examined, including the representation of states, events, and actions, as well as the handling of logical and natural language quantifiers, adverbials, modalities, and the meanings of complex concepts. In addition to examining these problems some concrete proposals are made to adequately depict their meaning structures.

A lexical structure is proposed that is especially well suited for the type of processing necessary to produce the meaning structures described herewith. Words convey a tremendous amount of information in communication and therefore their lexical meanings need to include a highly structured, information dense representation. At the same time the organisation of the lexicon must allow for swift relevant

retrieval and this has been accomplished.

Finally, methods by which the meaning structures are derived from text are outlined as well as methods for making some (common sense) inferences. Included as appendices are the results of a small implementation fragment of some of the ideas presented. This implementation is written in MACLISP and illustrates the processing of text from (unprocessed) English utterances through to the creation of semantic structures that represent the meaning content of the utterances.

ACKNOWLEDGEMENTS

I would like to thank Dr. Len Schubert, my supervisor, for his guidance, inspiration, creativity, and example. The pride I take in this thesis is due in large part to him. I thank Dr. Kelly Wilson and Dr. Jeff Sampson who have read and commented on everything I have written and for serving on my committee. I am grateful for the careful reading and insights of my external examiners, Dr. Lois Marckworth of the Department of Linguistics and Dr. John Mylopoulos of the University of Toronto. Fellow graduate students have also supplied me with fruitful discussions and ideas, especially Ian McMaster and Rici Liknaitzky. Rici also helped by programming some of the lexical maintenance routines used in the experimental program.

I gratefully acknowledge the financial assistance provided by the National Research Council of Canada and the University of Alberta, in the form of scholarships, and the Department of Computing Science in the form of teaching assistantships.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Focus

> "Would you tell me, please, which way I ought to go from here?"
> "That depends a good deal on where you want to get to," said the Cat.
> "I don't much care where-" said Alice.
> "Then it doesn't matter which way you go," said the Cat.[1]

Man is a theorist. The nature of his theories reflects the questions he wants to answer. This research is directed primarily at answering part of the following question: How can computers be made to understand natural language?

A long range goal of having each person able to communicate with a computer in the same way he communicates with other persons (viz. through natural language) cannot be achieved without significant applications of, and advances in, artificial intelligence techniques. What is needed is a general theory of natural language understanding to form the basis for computer programs that actually do understand. As such, this thesis describes theory and not a computer program although I have written some experimental programs (see Appendices).

---

[1] from Alice's Adventures in Wonderland by Lewis Carroll.

The theory put forth in this thesis is by no means complete. Not surprisingly, complete natural language understanding theories have proved elusive and yet, some remarkable theories that treat a wide range of problems have been postulated.[2] This type of work is cited throughout this thesis and has influenced my thinking immeasurably. Nevertheless, no theory explicates adequate methods for automating natural language understanding, or accounts for the way in which people understand (ordinary) language.

Early reflection on philosophical issues has convinced me that extensions to existing theories would fail in some instances or at best require messy, "special case" handling. Thus I have attempted to investigate problems that have defied systematic solution in any one theory and to account for some of them. Some of these problems have been solved for special environments; others, not at all.

The problems I was originally most concerned with led to a detailed consideration of the representation of states, events, actions, cases, causes, and intentions, as well as logical and natural language quantifiers, adverbials, relative terms, vagueness and uncertainty, the meanings of complex concepts, and time.

---

[2] See Schank (1972, 1973b), Wilks (1973a, 1973b, 1973c), Winograd (1972), Montague (1969, 1972), Lewis (1972), and Lakoff (1972).

The next section contains a discussion of some closely related approaches to this type of research. In the second chapter, some of the early notions that have helped shape the remainder of this research are discussed. The middle four chapters explain the components of state-based conceptual theory. In Chapter 3 the expressive power of semantic network notation is extended to account for such necessary constructions as quantifiers, logical connectives, and descriptions. These ideas derive from the paper by Schubert (1974). In Chapter 4 a detailed discussion of states and events, and the meanings of complex concepts is presented and further illustrations of the uses of the semantic network are shown. Memory and the lexical structure are developed and explained in Chapter 5. Particular attention should be paid to the lexical structure. I contend that words convey a tremendous amount of information in communication and therefore their lexical structure needs to include a highly structured, information dense representation. At the same time lexical organisation must allow for swift relevant retrieval and this has been accomplished. Finally, Chapter 6 contains a description of the methods through which meaning structures are derived from (unprocessed) English text as well as methods for making some (common sense) inferences. The final Chapter contains concluding remarks.

1.2 Related Work

Simmons (1968, 1970) has written two excellent review articles surveying relevant question-answering literature.[3] The first article reports on the early work on question-answering systems that were constructed along the lines of list structured, text based, and logical inference systems. The list structured work includes the SAD SAM system of Lindsay (1963), the BASEBALL program of Green et al. (1963), and the DEACON breadboard of Thompson et al. (1964). PROTOSYNTHEX by Simmons and McConlogue (1964) is the most prominent of the text based systems. But perhaps the more important systems are the logical inference systems such as Raphael's (1968) SIR program, Bobrow's (1968) STUDENT program, Cooper's (1964) Fact Retrieval system, and the Specific Question-Answerer of Black (1968). In his second article, Simmons reiterates some of the research he reported on in the first article and includes some additional reviews. The most interesting of these include Quillian's (1968) Semantic Memory program, Charniak's (1969) CARPS system, Slagle's (1965) DEDUCOM system, and Weizenbaum's (1966) ELIZA program.

Since the time of Simmons' reviews a number of interesting research efforts have taken place. Notable systems developed from 1970 until 1973 deserve mention although they

---

[3] Question-Answering research appears to represent the most direct attack on programming a computer to pass Turing's test, see Turing (1950).

have not directly influenced my own research effort. This period includes the research reported by Woods (1970), Woods et al. (1972), Kaplan (1972), Bruce (1972) [CHRONOS], Coles (1972) [ENGLAW], and McCalla and Sampson (1972) [MUSE]. Because of Simmons' reviews and the large volume of related material, it is unnecessary to survey the field once more here. Rather I will point the interested reader to Simmons' reviews and to the original papers, and try to concentrate on research which has had more cogent influence on this work, i.e. Schank (1972), Wilks (1973), Anderson and Bower (1973), Rumelhart et al. (1972), Winograd (1972), and Charniak (1972).

Winograd's (1972) language program and Charniak's (1972) children's story model endorse the procedural meaning representation for knowledge due to Hewitt (1971). In this way factual and heuristic information are combined to promote more efficient (w.r.t. processing time) use of the factual information. Nevertheless, the visual suggestiveness inherent in network oriented representations, which leads to efficient processing algorithms, is lost (see Chapter 3 for a complete discussion of this point). More importantly, this approach to representation (the procedural embedding of knowledge) runs into difficulties when contexts shift. Different heuristics and processing methods are then appropriate.

Winograd's system combines many fascinating techniques to produce spectacular results. In his system, semantic analysis takes place before syntactic processing is complete.

If semantic analysis fails for the structure built thus far, a backtrack mechanism allows the parsing to be redirected along more promising lines. This was the first true integration of syntactic and semantic analyses with good results.

PROGRAMMAR, a language for writing grammars, was developed by Winograd in order to get systemic grammar into a usable form (for LISP programming). Winograd found systemic grammar more amenable for finding meaning related structural units in natural language as opposed to the traditional syntactic use of formal grammars, e.g. phrase structure and transformational grammars. Systemic grammar is a non-formal grammar that produces parse trees with few but complex nodes.

Criticisms that might be leveled at Winograd's system include the following. First, his semantics are tied to the simple referential blocks world without a method to make the semantics extensible to any general real world situation. For example, Winograd's system would be unable to correctly decide between alternative word meanings in any given context. It is just plain difficult to set up new "micro-worlds" in which understanding would take place, even if one were so inclined. Secondly, the use of MICROPLANNER as a deductive mechanism can be criticized because of MICROPLANNER's use of blind backtracking to handle errors (although it should be noted that no better working implementation of a deductive system was available at the time). Finally, the use of procedures as meaning representations is acceptable in some cases but

awkward in many cases where a "static" representation would be sufficient.

Charniak's model is aimed at answering questions about children's stories where a great deal of "common sense" knowledge is needed. In order to incorporate this kind of knowledge into his model, Charniak made extensive use of demons, which are antecedent theorems in the PLANNER sense. In this way, many assumptions can be filled in that are not explicit in the story. Invocation of demons occurs whenever appropriate patterns are matched in the input.

Charniak's program operates decoupled; this means that the initial "parsing" was effectively decoupled from the inferential work that Charniak thought more interesting, e.g. pronominal references, and simply assumed. However, the inferences required to resolve words' sense ambiguities and pronominal references are linguistically related. At times sense ambiguity and pronoun reference problems occur in the same sentence and must be resolved together. Decoupling seperates the process so presumably the word sense ambiguity must be done in the future by a different (or subsequently recoupled) system. Charniak's model, however, is still not fully developed and the recent attention focused on "frame" systems seems to follow more or less naturally from his effort in creating demons.

Rumelhart, Lindsay, and Norman (1972) have developed a

model for long term memory that was to be

> ... capable of encoding and representing a reasonable range of the information it is likely to encounter, that has direct and explicit rules for translating external information into an internal representation, and that is flexible enough to support a variety of cognitive and information-processing tasks.

That they have succeeded to some extent is evidenced by a recent thesis by Scragg (1974) under their tutelage. Scragg developed LUIGI, a system that answers questions about processes and makes use of the model and representation developed by his mentors. I will discuss their work in the context of Scragg's thesis since it is more recent. Though I initially found the model appealing, their model includes a number of inconsistencies.

The introduction of the binary predicates ISA and HAS-PROPERTY (which can be thought of as applying their second arguments to their first arguments) is a trick used by advocates of semantic networks to artifically convert unary to binary predicates. This is pointed out by Schubert (1974) as an unnecessary contrivance unless one's purpose is to convert a higher-order logic into a many-sorted first order logic (see Chapter 3).

In his thesis, Scragg gives the following diagram for the memory structure generated by LUIGI for a cheese sandwich.

```
<*19>---------->sandwich
  |   superset
  |
  | ingredient
  |
cheese
```

This leads to immediate difficulties since he appears to use nodes like <*19> as both specific elements and as generic nodes. If <*19> can have a superset, then it is a set; but sets can't have ingredients, only their members can! Then the question can be posed: is this *some* set of cheese sandwiches, or the set of *all* cheese sandwiches? Note also that the *concept* cheese cannot be an ingredient.

Other difficulties with the formalism developed by Rumelhart et al. (1972) center around adverbial modification. Their approach is to treat adverbial modifiers as operators that apply to relations and other operators to generate new relations or operators. Unfortunately, many adverbial modifiers require systematic analysis rather that mere replacement by "n-valued" relations as Rumelhart et al. do (see section 4.7). Despite the theoretical inadequacies mentioned, Scragg has managed to construct a system with relatively good performance.

Human Associative Memory [HAM] by Anderson and Bower (1973) is mainly directed toward attaining an adequate theoretical framework for describing comprehension, for exhibiting the internal representations of propositional information, and for characterising the "interpretative

processes" that encode this information into memory: in short, developing a theory about human memory.

In their representation, sentential forms and predicates are decomposed into binary trees with labeled branches. While I see nothing intrinsically wrong with this type of representation, the significance attached to link labels and to the extra nodes they introduce is puzzling. If the labels are mnemonic distinguishing marks possibly indicative of more complex relations at some more detailed level of analysis, then I see no problem. The extra nodes seem to be regarded as arguments of further propositions and adverbial modification is one instance where they use the extra nodes for making further propositions. However, since they are tied ultimately to relations at the bottom of a propositional tree via set membership, they presumably denote single n-tuples. Nevertheless I cannot interpret adverbial modification of an n-tuple.

Although a number of other issues concerning Anderson and Bower's representation are troubling, they at least attempted to include into their notation provisions for handling quantification and implication. However, I do not find their methods adequate as they have problems with quantifier scopes; also the use of set-subset relations as giving the full power of implication may lead to troubles, especially with quantified implicative propositions with complex antecedents. I don't see how they distinguish "the set

of all dogs that chase cats" from "a set of dogs that chase cats."

Schank's (1972) conceptual dependency theory is the basis of a system developed over about six years culminating in the MARGIE system by Schank et al. (1973). It is a system rich in semantic representation designed to provide a representation of meaning in terms of which paraphrase, inference, and machine translation could be carried out.

Schank has succeeded to some extent with the first of these two aims. MARGIE operates in two modes: the paraphrase mode and the inference mode. Following is a sample output from those two modes.

```
INFERENCE MODE

     INPUT:  John told Mary that Bill wants a book.
     OUTPUT:  A book about what?
              Mary knows that Bill wants a book.
              Bill wants to come to have a book.
              Bill wants someone to cease to have a book.
              Bill wants to read a book.

PARAPHRASE MODE

     INPUT:  John killed Mary by choking Mary.
     OUTPUT:  John strangled Mary.
              John choked Mary and she died because she
                could not breathe.
              Mary died because she was unable to inhale
                some air and she was unable to inhale some
                air because John grabbed her neck.
```

Schank uses a graphical notation consisting of items from four conceptual categories to represent meaning structures. These categories are picture producers [PP], picture aiders [PA], action aiders [AA], and actions [ACTS] and they correspond

closely to nouns, adjectives, adverbs, and verbs, respectively. The smallest structural unit that Schank deals with is the _conceptualisation_. Conceptualisations are graphical structures that link together conceptual categories using a variety of graphical symbols, conceptual tense markers, conceptual cases (analogous to and certainly influenced by the linguistic case structure of Fillmore, 1968), and primitive actions.

Schank currently uses four cases which serve as subgraphs in conceptualisations. These cases include the _objective_ case, which relates an objective PP to an ACT; the _recipient_ case, which relates a donor PP and a recipient PP to an ACT; the _directive_ case, which relates direction (to and from) to an ACT; and the _instrumental_ case, which links conceptualisations instrumental to an ACT to a conceptualisation containing the ACT.

In addition to conceptual cases, Schank makes use of only fourteen primitive actions through which he expresses all other actions. These primitive actions are: PROPEL, MOVE, INGEST, EXPEL, GRASP, PTRANS, MTRANS, ATRANS, SMELL, LOOK-AT, LISTEN-TO, CONC, and MBUILD.

I wish to defer a further discussion of Schank's system at this time. It is sufficient to say that I consider it one of the very few natural language processing systems that may truly fall into the category of _understanding_ systems. The

final system I wish to discuss might also fall into this category, i.e. Wilks' (1973) _preference semantics_ system.

Wilks' system, like Schank's has a uniform representation in terms of structures of primitives for representing the meaning content of natural language. Unlike Schank, Wilks has concentrated on machine translation, from English to French, of small input paragraphs and he has reported reasonably good translation. His system does not operate in paraphrase or inference modes.

Wilks' system makes use of _formulas_, one for each meaning sense of a word. These formulas are based on the (binary) decomposition trees developed by Lakoff (1972). The formula is a tree structure of semantic primitives interpreted formally using dependency relations. A typical formula for the action of "drinking" is as follows

((ANI SUBJ)((FLOW STUFF) OBJ)(SELF IN)((THIS(ANI(THRU PART)))
TO)(MOVE CAUSE))

The rightmost element is called the _head_. Template structures, that actually represent sentences, are built up as networks of formulas. These templates always consist of an agent node, an action node, and an object node, as well as any other nodes that may depend on these three formulas. Formulas dictate how other places in the template should be filled. Thus "drink" would _prefer_ a FLOW STUFF as object and an ANIM as subject. Prefer is the correct word to use since if either a non ANIM subject or a non FLOW STUFF object are the only choices

available, the utterance will still be recognised (metaphorically). The template finally established for a fragment of text is the one in which most formulas have their preferences satisfied. This very simple device is able to do most of the work of a syntax and word sense ambiguity resolving program.

After the agent-action-object templates have been set up locally for fragments of input text, Wilks' system attempts to tie these templates together to provide an overall meaning structure for the input. To accomplish this, Wilks makes use of <u>paraplates</u> attached to formulas for English prepositions. These paraplates range across two, not necessarily contiguous, templates.

Thus far the structure of mutually connected templates comprises a semantic block. This is all done in what Wilks terms his <u>basic</u> mode. Whenever sentences cannot be successfully resolved into a semantic block in the basic mode, Wilks employs another mode, the <u>extended</u> mode, which makes use of common sense inference rules. This mode attempts, by a simple strategy, to construct the shortest possible chain of rule-linked template forms from previous text containing one of its possible referents. This chain then represents the solution to the ambiguity problem.

After constructing a semantic block, French generation proceeds by "unwrapping" the block. There is no deepening of

the representation by the generation routines.

The conceptual dependency approach of Schank (1972, 1973a, 1973b) and Schank et al. (1973) and the preference semantics approach of Wilks (1973a, 1973b, 1973c) exemplify what I believe to be the correct approach to the problem of representing the conceptual content of natural language utterances in terms of meaning structures. Specific criticisms of their respective approaches have been given elsewhere, see Cercone and Schubert (1974). Nevertheless, these two related approaches have the following desirable features regarding knowledge representation.[4]

(i) Interpretive directness

The meaning structures corresponding to natural language utterances are formed according to simple structural rules. Powerful heuristic criteria, based on the central role of verbs and on preferred semantic categories for the subjects and objects of verbs, guide each choice in the creation of meaning structures. Interpretation of utterances then takes on a "slot and filler" character, rather than requiring extensive trial and error search.

(ii) De-emphasis of syntax

In ordinary discourse it would be absurd not to accept "ungrammatical" constructions like dangling participles or fanciful locutions such as metaphor. Neither preference semantics nor conceptual dependency imposes a syntactic straightjacket on admissible utterances. Therefore the abnormal is not excluded as it is in many linguistic systems.[5]

---

[4] These features have not all been articulated by either Schank or Wilks and they would perhaps dispute this characterisation of their respective approaches.
[5] Cf. the selectional restrictions of Katz (1964).

(iii) Emphasis on events

A major part of our interpretative effort in understanding natural language is focused on events, i.e., time-dependent relationships. By contrast, "static" relationships in the world are relatively easy to understand. Therefore the search for fundamental semantic structures and primitives should concentrate on the representation of events.

(iv) Focus on actions

Both Schank and Wilks have shown that there does exist a small, more or less adequate, set of actions through which a surprisingly large number of action concepts can be expressed. Using this minimal set of actions it is relatively easy to use the meaning representations in a language (and paraphrase) independent way. Relatively few inference mechanisms will then be required.

(v) Anaphoric disambiguation

An important by-product of the notion of semantic preference, used in setting up meaning representations, is the disambiguation of certain classes of anaphoric reference. An example of one type of anaphora that is directly accommodated by Wilks' representation (a denser network of links accompanies the correct referent) is given by the sentence: "Often union workers go on strike knowing they are not good for the economy. Most probably they need the money."

Specific difficulties with Schank's and Wilks' approaches are discussed and plausible ways of overcoming some of their difficulties suggested beginning in Chapter 3 and continuing through Chapter 4. The next chapter discusses some thoughts on knowledge and representation.

Chapter 2

KNOWLEDGE AND REPRESENTATION

2.1 Uses of Knowledge

Knowledge is possessed by data processing systems in the form of internal representations. Usually knowledge is organised and represented in ways that are appropriate for the various uses that the processing system must make of the knowledge. For example, the nature and variety of available representations affects problem solving ability. Often in working with a difficult integral a human problem solver might shift representations several times, say from trigonometric functions to rational functions and so on, to develop a solution. A computer scientist might benefit from knowing about multiple representations such as a ring structure for storing sparse multidimensional arrays when storage is at a premium and linear non-linked structures when execution time is of paramount importance.

Language processing systems make several prominent (and related) uses of knowledge. Initially, in dialog situations, these systems must represent the meaning content of utterances for the purpose of comprehension. Often, in setting up these meaning representations for utterances, inferences must be drawn to account for missing or incomplete information to

achieve the representation. Later, other inferences may be drawn to answer questions. These inferences are drawn from general knowledge that is entered *a priori* to the processing system, as well as knowledge conveyed through the dialog. The representation of such knowledge, in a consistent and readily available form, is a central problem in artificial intelligence research and is discussed further in the next section.

2.2 Knowledge Representations

Some of the main approaches to representing knowledge in natural language utterances have used semantic networks (Quillian, 1968, 1969; Schank, 1972; Anderson and Bower, 1973), logical statements (Sandewall, 1971; Coles, 1972; Moore and Newell, 1973), procedures, (Winograd, 1972; Hewitt, 1971), and descriptions, (Moore, 1973).

The predicate calculus or logical statement approach represents knowledge in the form of propositions. Conjunctive normal form is often used where each proposition is a disjunction of a set of literals. Literals are atomic formulae composed of predicates and terms where predicates are functions of variables over a domain of discourse. One big advantage of the predicate calculus approach is having a uniform representation for all knowledge. This tends to make the control structure for programs manipulating knowledge

quite simple, but the benefits of multiple representations are ignored. Using only first order predicate calculus, the expressive power is sufficient to represent almost anything that we can formulate precisely (e.g. most of mathematics). However, many problems can be conveniently and compactly represented by higher than first-order logic and by modal logic.

Knowledge can also be stored in the form of programs rather than in the form of propositions. The knowledge embedded in procedures can be accessed using the new technique of pattern directed procedure invocation (see Hewitt, 1971). Advocates of knowledge representation through procedural embedding stress that the kinship between the way a person uses the knowledge he possesses for the variety of activities that he performs and his intelligent behaviour is due in large part to the "procedures" that he possesses for carrying on his activities. The use of procedures for representing knowledge, however, is very application dependent, e.g. PLANNER data bases (see Winograd, 1972). My concern is for a more general form for representing knowledge, application independent, that can make use of superimposed procedures as a heuristic device for a particular task.

Attention needs to be focused on a number of issues regarding representation. The first issue is that of scope (what kind of knowledge can be represented) as discussed in Moore and Newell (1973). The scope of representation that is

needed is one that accomodates the richness of language, i.e. the enormous variety of ideas that we can express.

In this thesis only this first issue (scope) regarding knowledge representation is attacked. The representation of natural language utterances is accomplished using a semantic network notation with extended expressive power sufficient to represent most utterances. Other issues concerning knowledge representation will be briefly discussed now but no definite solutions will be proposed to handle them.

A second issue regarding knowledge representation deals with the acquisition of new knowledge. Recent work by McDermott (1974) has focused on some aspects of this problem. Specifically, McDermott is concerned with adding factual knowledge to his data base in a manner that maintains integrity in the data base in much the same way people do, i.e. by altering belief structures.[6] Others, including Winograd (1972) have been more than marginally concerned with this type of problem. Major work needs to be done; this whole issue of knowledge acquisition is open for further investigation. To the best of my knowledge, no method exists for systematically acquiring knowledge in a general way.

A final issue regarding represention is that of multiple

---

[6] His data base is procedure oriented in the same fashion as Winograd (1972), with Conniver (see McDermott and Sussman, 1972) replacing Microplanner (see Sussman et al, 1971).

representations. The way we choose to represent a problem often influences the way in which we formulate a solution. Finding the appropriate representation, or modifying the existing representation, is an important activity. Having experience with a particular class of problems would help. In doing so we are dealing with multiple representations, which, thus far in Artificial Intelligence systems, have never been incorporated within the same system.

2.3 Problems in Representing Knowledge

Quantification presents a number of interesting representational problems. Consider the sentence

"All people have some bad habits." <2.1>

First of all the sentence is ambiguous. Two possible meanings include:

"Each person has a particular set of bad habits and that set may differ from person to person."; and <2.2a>

"There is a particular set of bad habits that all people possess." <2.2b>

Secondly it is doubly quantified. Johnson-Laird(1969) conducted experiments suggesting that doubly quantified sentences were predominantly interpreted with the greater scope belonging to the quantifier on the surface subject. Thus we would assign the sentence the meaning expressed by <2.2a>. If we accept this for the moment, the problem remaining is how to represent the meaning. The scope (logic definition) of

universal and existential quantifiers which present little difficulty in predicate calculus representation are problematic in semantic network representations. A convention that could be adopted would include the scope of any quantifiers in a predicate within the scope of the subject quantifiers. This is in fact done by Anderson and Bower(1973). The scope problem, however, doesn't disappear. Consider the following pair of sentences:

"There is someone who is always there." <2.3>

"There is always someone there." <2.4>

Here the surface subject is the same for both sentences while the existential and universal quantifiers are interchanged before the predicate. Quantification is a central problem in representation and is one that is considered at length in sections 3.2.3 and 4.2.

Certain other problematic constructions appear in ordinary discourse. Propositional attitudes and other modalities present special difficulties. For example, it is common to talk about certain information without explicitly stating that information. The utterance "I know John's phone number" is a problem in dealing with propositional attitudes. This sentence can be paraphrased as "There exists an x such that I know that x is John's phone number". This is an example of a transparent modal context, i.e. we're quantifying <u>into</u> the scope of "know".

Problems also arise with referentially opaque contexts.

The sentence

"John wants to marry the prettiest girl." <2.17>

gives rise to two possible interpretations (readings):

"John wants to marry a specific girl who also happens to be the prettiest girl."; and <2.18a>

"John has no particular girl in mind, but he wants whoever he does marry to be the prettiest girl." <2.18b>

The transparent reading is given by <2.18a> and the opaque by <2.18b>. Section 4.8 deals with referential opacity.

We often use counterfactual conditional statements in discussions. It is desirable to represent conditionals in a consistent manner; a manner in which they are not confused with events that actually occur. In chapters 3 and 4, these and other representational problems are closely considered.

2.4 Some Thoughts about Organising Knowledge

People working in information retrieval have long recognised the need to reorganise the data base when the cost of retrieving relevant data items becomes expensive. One of the main criteria in the design of a data base concerns the nature of the retrieval requests and the nature of the particular body of information to be organised. Once these issues have been settled, the design of the data base can proceed.

What, then, is required of a memory system for computer understanding of natural language? This kind of memory system

is to be used in making logical deductions and common sense inferences, in solving problems and formulating plans, in understanding verbal explanations of complex concepts, and in recreating past events for expository purposes. Some thoughts concerning this type of memory requirement are examined in the next two sections.

2.4.1 Some Notions About Memory

Many diverse groups have theorised about the structure of memory. Philosophers, linguists, psychologists, and computer scientists are the most prominent of these groups. Understandably, this has led to differences in the terminology used among various authors writing on the subject. For that reason, it is desirable to explain some of the terminology used herein.

Memory is viewed as that part of an information processing system that can selectively retain information. The term semantic memory refers to the type of organisation of stored facts (namely, networks), and the routines used to process these facts. Lexical memory represents the knowledge that people possess about words and other verbal symbols, the meanings of words and their referents, relations among words, as well as rules and algorithms for manipulating the symbols and relations.

A function of semantic memory is the ability to retrieve

information that was not explicitly stored in memory.[7] Information stored in semantic memory represents objects, concepts, relations, states, events, facts, propositions, and so on.

The structure of semantic memory refers to the way in which information is represented in memory. This structure requires a formalism with sufficient expressive power to represent natural language. Predicate calculus is one formalism with the necessary expressive power but suffers from major (methodological) disadvantages that will be discussed in Chapter 3. The semantic memory representation used in this research is referred to as a semantic_network. It remains to be shown (Chapter 3) that semantic networks have the expressive power of predicate calculus; however, they do have the advantage of visual suggestiveness that is unattainable in predicate calculus. Whereas predicate calculus representations encourage the application of deductive inference techniques, semantic networks are suggestive of the type of cognitive processes humans are capable of when they use language, e.g. association.

---

[7] One standard way to do this is to semantically interpret the retrieval request by expanding the relational and predicate components of the request. For example, one might be asked to list the literature one has read. This request could initially fail if the information was stored as novel, plays, and poems. Thus the term literature could be semantically expanded so the request is reformulated to ask for a list of the novels, plays, and poems one has read.

In actual implementation, semantic information is normally represented in a computer as property lists attached to concepts. In turn, elements of property lists may be pointers to other lists. A concept then could be represented as a set of interrelationships among other concepts. A memory structured in this way becomes a network of interrelated lists with pointers attached to words on many of the lists. Word senses can be represented unambiguously since any word with more than one meaning would have more than one list attached. Since property lists are indefinitely expandable, no restriction is placed on the level of detail of representation.

The description just given is very superficial. Without considering the relationships between properties and concepts, we are left with just an associative network. We need to explore the nature of concepts a bit more closely to see what we are dealing with in this associative-type network.

The relationships between concepts can be as different as the concepts themselves and in fact these relationships are concepts. Collins and Quillian (1972) make the point

> ... green is related to grass in a different way than yellow is to canary, since green penetrates the grass and the yellow is only superficial to the canary. Both relations are different from the relation of blue to sky, since the blue is only in the atmosphere of earth during the day. These examples illustrate that relations can be quite complex, even though the question of "What color is grass?" can be asnwered without getting into these complexities. Any representation of relations in a computer must permit them to be as detailed as

necessary; in other words, the description of a relation must be embeddable.

It might seem reasonable to assume that concepts can be organised hierarchically with general concepts towards the top and narrow concepts towards the bottom of the hierarchy. Certainly there is evidence of this type of organisation in humans. Consider the taxonomy of the plant or animal world. However, many relationships exist that are not strictly hierarchical, e.g. certain geometrical relations, causal relations, precedence relations, etc. Thus I am not suggesting that a hierarchical organisation is proper for all concepts but rather that it can help in developing heuristic memory-organising routines.

To allow for inference, we need to consider the semantic organisation of concepts. While all relations allow for inferences, classes of inferences are specific to particular relations. Underlying the hierarchical organisation is the subconcept-superconcept relation. Collins and Quillian (1972) suggest that frequently questions that could be answered easily on the basis of explicitly stored propositions are nevertheless answered through inferences facilitated by a hierarchical organisation. It is doubtful that a person would store the fact that a rose is a plant but rather that it is a flower and infer that it is a plant.

We gain further insight into memory organisation when we consider how we might use memory elements for language

comprehension. Comprehension involves selective retrieval. When a person understands an utterance, in effect he has found an interpretation with respect to the knowledge that he possesses that makes sense of the utterance. We do this all the time when we reason by analogy, speak metaphorically, or resolve anaphoric references. The sentence "The rain drums on the shelter" should probably be interpreted something like "The rain falls on the shelter and produces a hollow sound like someone beating on a drum." Quillian (1968,1969) has developed some strategies for the type of search needed here. His intersection technique systematically proceeds outward along links from concepts denoted by words in the utterance. It is an ordered serial search that looks for all possible intersections between the designated concepts. All connections found are checked to see if the relations between the concepts meet syntactic and contextual constraints. Thus in the example above, if the utterance is to be understood, the search must yield an intersection between the concepts "rain" and "drums".

Adding information to memory remains problematic. Further complications ensue since most properties of concepts are not directly communicated to an understander but rather are abstracted in some manner. This "naturally" human process is not one that is readily mechanised with existing techniques and thus requires considerable thought and further investigation.

2.4.2 Some Notions About Lexicons

Language use presumes knowledge about words. Part of this knowledge is functional and part of it deals with the meanings of words. The lexical component of memory should be organised adequately to deal with both.

As a starting point in investigation, we might consider how any standard dictionary is organised. This consideration quickly leads to chagrin. Words are not assigned meanings in any precise way; rather, various methods are used, be they extralinguistic (e.g. diagrams) or explicit definitions. One problem is that meanings are given in English, which is not a suitable memory formalism. This informal structure is not conducive to the construction of an *in toto* memory model. What, then, are the important considerations?

The most important practical concern is the storage of lexical items. For storage economy, the lexicon should be generative in character. The structure should be such that most relevant meaning of words be inferred and little information about the word be directly stored. One obvious example of this would include a method of performing morphological analysis on words. This facility would permit a single lexical entry to be made for the root form of words along with an indicator for permissible inflections (see Cercone, 1974, and also Appendix A).

In these first two chapters we have set the theme for the rest of this thesis by focusing attention on the problem of representing natural language. Several earlier approaches to this problem were examined and some problems with the use and representation of knowledge for language processing were discussed. In chapter 3 the notational basis is established for a state-based conceptual representation for natural language utterances. In that chapter the expressive power of semantic networks is systematically extended in order to account for many types of natural language constructions. Subsequent chapters use the extended network notation to express states and events, the meaning of complex concepts, and the meanings of word concepts. Finally, an experimental program that makes use of these concepts is described and many examples are given that illustrate the kinds of meaning structures that the program builds.

Chapter 3

SEMANTIC NETWORKS

3.1 The Expressive Adequacy of Semantic Networks

Semantic networks appear to be in vogue presently. Yet there seem to be some misconceptions concerning their use in representing knowledge. Most prevalent of these notions are statements of the form: semantic networks are more semantic than the syntactically oriented predicate calculus approaches to knowledge representation. But obviously both are formal languages designed to allow natural language statements to be paraphrased precisely and unambiguously.

The respective reputations of semantic network representations and predicate calculus representations can be attributed to the use to which they have been put. Research into natural language processing based on semantic networks has emphasized associative and other nondeductive processes and has had significant results. Predicate calculus has usually been wedded to resolution-based theorem proving. However, one could just as easily adapt theorem proving techniques to the semantic network representation or design nondeductive inference algorithms for predicate calculus.

Schubert (1974) extended the expressive power of semantic networks by systematically incorporating logical

connectives, quantifiers, descriptions, and other constructions. This extension further closes the gap between semantic networks and predicate calculus.

The motivation to extend the expressive power of semantic networks arose from the methodological disadvantages inherent in linear representations, such as the predicate calculus, for representing knowledge in natural language understanding systems. Semantic networks provide readability; they are suggestive of appropriate data structures as well as procedures for inference and interpretation. These advantages are none the less important for their lack of theoretical import. "Associative processing" of the type done by Quillian (1968) and the algorithm for matching complex scene descriptions of Winston (1970) used a semantic network notation. It seems unlikely that either would have accomplished his respective task without the use of semantic networks even though in both cases the networks are reducible to sets of propositions based on binary relations.

The following sections incorporate into semantic networks the notation required to attain the increased expressive power that is necessary to represent many natural language utterances in a logically adequate way. The extended notation can apparently express anything that can be expressed in English, preserving distinctions between utterances with distinct meanings and between distinct readings of ambiguous utterances. Furthermore, judgments of truth and falsity about

English utterances appear to correspond appropriately to truth and falsity in the formal representation.

3.2 Representing Knowledge in Semantic Networks

This section contains a rough paraphrase of an earlier report by Schubert (1974). That report established a notational basis for concurrent research of the "state-based" conceptual representation by L. K. Schubert and myself. While retaining the central ideas of the original paper, some of the detail has been omitted.

Semantic networks are used in this thesis as graphical analogues of data structures that represent facts in a computer system for understanding natural language. Many artificial intelligence researchers have used semantic nets in their respective analyses. In doing so they have made use of the following characteristics of semantic networks. First (and most important), nodes that denote the same concept are not duplicated (in most cases). It is then possible (and frequent) that distinct propositions may impinge on a node via arcs. Second, propositions are formed by linking predicate names to their argument nodes using arcs. Third, since concepts are not necessarily word concepts (need not have names attached), particular and general concepts are represented as labeled or unlabeled nodes of a graph.

All semantic networks proposed thus far, including those

of Quillian (1968), Winston (1970), and Rumelhart et al. (1972), can readily be translated into a linear notation whereas linear notations can only be trivially translated into a tree or network notation. If only for the methodological advantages then, it is apropos to extend the expressive power of semantic networks to enable translation from linear representations into semantic networks.

The remainder of this chapter deals first with some introductory notation for semantic networks and secondly with logical deficiencies and their remedies.

3.2.1 Introductory Notation

In semantic net notation, the distinction between labels designating storage locations and labels designating pointers to storage locations requires clarification. This distinction is used by Quillian (1968) to designate "type nodes" (unique storage locations) versus "token nodes". The notation can be made uniformly explicit as shown in Figure 3.1.

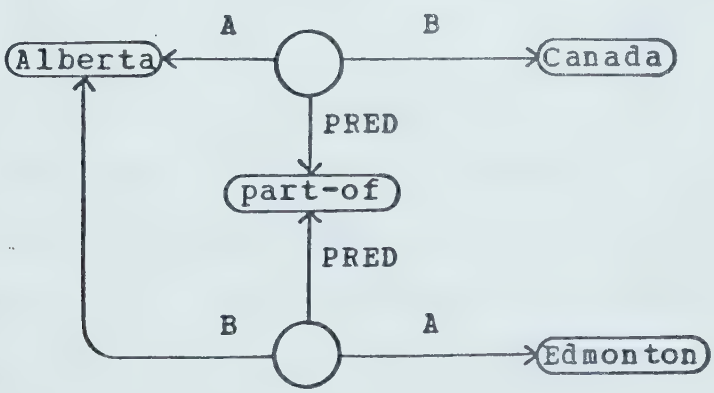


Fig. 3.1 "Alberta is part of Canada. Edmonton is part of Alberta."

Here "part-of", which in some notations corresponds to a token node, designates a type node. All encircled nodes correspond to storage locations and all arrows to addresses of storage locations. What formerly were token nodes are now called proposition nodes; they serve as graphical nuclei for propositions as a whole. Nodes may be labeled with names for the concepts they denote, e.g. John, book, book1, book2; ordinary attributive terms such as "book" are reserved for the corresponding universal concepts, while numerically suffixed words such as "book1" are used for particular instances of the concepts.

At times the explicit notation of Figure 3.1 will clutter the diagram leading to a loss in readability. Therefore, when the meaning is clear, binary predicates will be represented as in Figure 3.2 for visual effect with the understanding that the use of explicit propositions underlie the structure.

Alberta ——part-of——> Canada

Fig. 3.2 "Alberta is part of Canada"

In Figure 3.1, A, B, and PRED are mere distinguishing marks. They are analogous to parentheses or commas in the predicate calculus in that they serve to relate denoting terms syntactically; they are non-denotative themselves. Whenever possible they will be chosen to be meaningful, i.e. to enhance readability and be suggestive, but they could be chosen as

numeric labels as well.

One advantage of the explicit notation of Figure 3.1 is that it works for n-ary (n>2) predicates. The sentence "John gives the book to Mary" involves "gives" as a three place predicate.[8] It is diagrammed as in Figure 3.3.

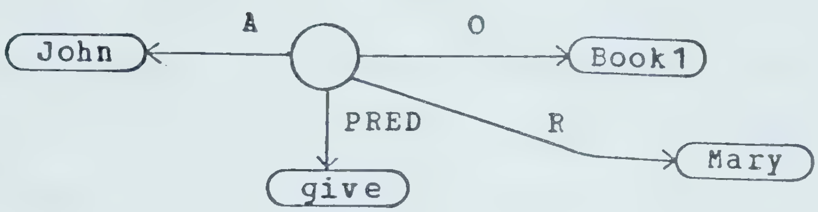


Fig. 3.3 "John gives the book to Mary"

Figure 3.3 is appealing because of the significance we can attach to labels - agent, object, and recipient. By no means is Figure 3.3 a graphical analogue of "case-structured" grammars. Cases are not viewed as <u>conceptually primitive binary relations</u> as Fillmore (1968) and researchers influenced by him, notably Schank (1972), view them. In a case structured system the central node would denote a specific action or process with the property that it is a "giving" and involves John, the book, and Mary as agent, object, and recipient respectively. A discussion of "case" appears in section 4.2. The "case" labels in Figure 3.3 are to be regarded as mere mnemonics, although indicative of more complex relations. To

[8] In Winograd's (1972) work, "gives" is recognized as a transitive action that requires two objects: his classification is TRANS2.

avoid confusion, predicate names will be designated in small letters and markers by capitals.

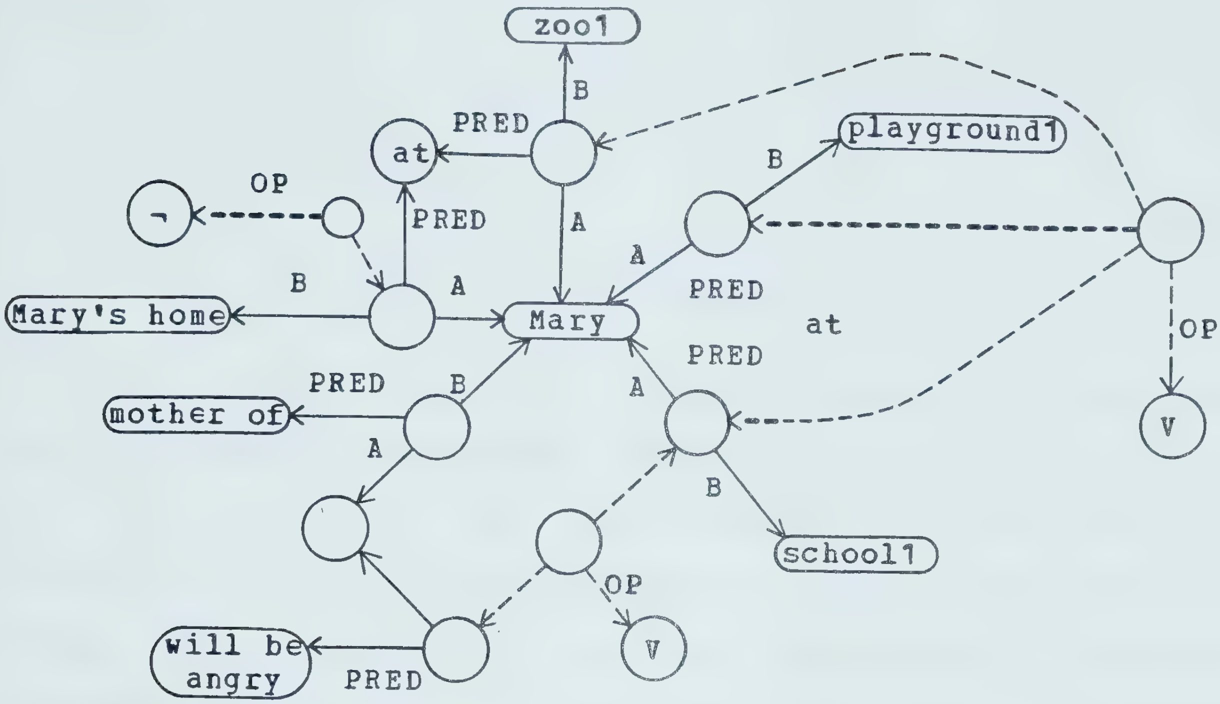
3.2.2 Logical Connectives

In most semantic network formalisms, scant use is made of logical connectives. Normally all propositions in a semantic network are assumed asserted and thus their conjunct is asserted. Nevertheless disjunction and other connectives are frequently used in discourse and are necessary for truth-functional completeness. The introduction of explicit nodes for logical compounds of propositions (or open sentences), with graphical links to the components, solves this problem. Figure 3.4 illustrates the formation of disjunctions by the use of graphical links to tokens of the disjunction operator. Figure 3.5 shows the general form of the disjunction operator.

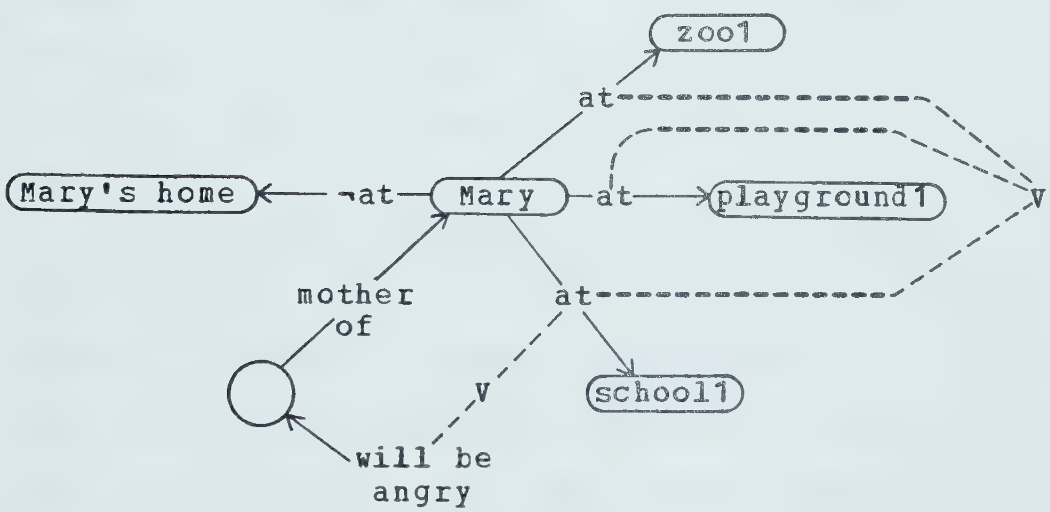
In Figure 3.4 _broken lines_ are used for the operator-operand links of the logical operator. Note that no distinguishing marks are needed on the links (disjunction is a symmetrical operator) and arrowheads can be dropped when there is no ambiguity. The use of "will be" as a modifier of "angry" is an evasive manoeuvre; it postpones discussion of time.

If desired, other logical connectives can be introduced in exactly the same way. For example, it would have been more natural to render "If Mary is not at school her mother will be angry" by means of implication (ignoring the implicit causal

proposition) instead of disjunction.



(a) Full Notation



(b) Abbreviated Notation

Fig. 3.4 "Mary is not at home; she is either at school, or on the playground, or at the zoo; if she is not at school, her mother will be angry."

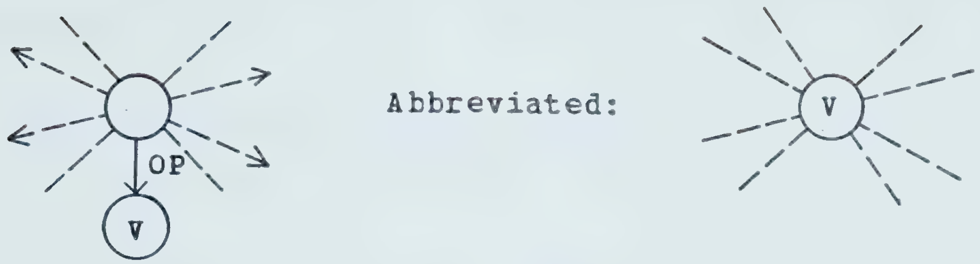
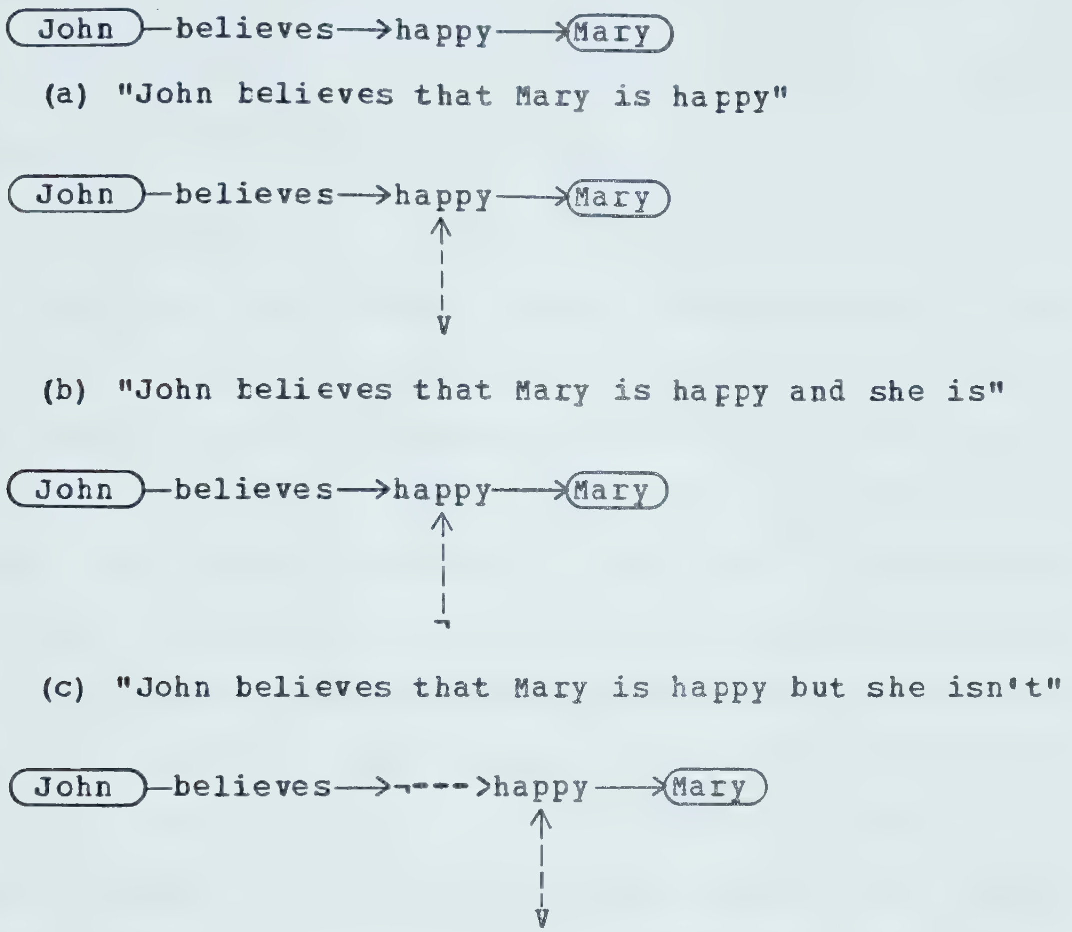


Fig. 3.5 Generalized disjunction.

In a semantic network containing logical compounds, the usual convention of regarding all propositions in the network as asserted doesn't work. In this formalism the convention adopted is that the complete semantic net <u>asserts exactly those propositions which are not constituents of compound propositions</u> (i.e., operands of connectives or modal operators). Graphically this means that exactly those propositions are asserted that are not pointed to (thus in Figure 3.4, "Mary is not at home" and "Mary is at school or her mother will be angry" are asserted whereas "Mary is at the zoo" and "Mary is at school" are not).

This raises the question of how to assert a proposition which is also a constituent of a compound proposition. For any logical compound, simplifications result if a constituent is asserted. For propositional attitudes, causes, intentions and the like, however, it may prove worthwhile to assert a proposition independently of the compound. In this case, we can use disjunction with a single operand, V --> p, as a way of saying "p holds" since the compound proposition established by the token V is not pointed to. Further examples are shown

in Figure 3.6 (involving modal operators that are discussed in section 3.2.7).



(a) "John believes that Mary is happy"

(b) "John believes that Mary is happy and she is"

(c) "John believes that Mary is happy but she isn't"

(d) "John believes that Mary isn't happy but she is"

Fig. 3.6 Asserting propositions by means of monadic disjunction

3.2.3 Quantifiers

The treatment of quantifiers in semantic networks has generally been rather cursory. Often quantifiers are regarded as monadic modifiers of concept nodes, indicative of "how many there are" of that item (i.e., set cardinality). Universal

quantifiers are then attached in the same way, even though the logical operator A ("for all") is not at all indicative of cardinality over the set whose members are quantified. Rather, as discussed later in this section, universal quantifiers <u>relate</u> the cardinality of two sets.

Quantification needs to be included in any representation for natural language understanding for several reasons. First of all, logical and natural language quantifiers appear in discourse, for example "John sent the cards to all of his friends" and "Several of my friends were at the game today". Secondly, the use of quantifiers is required in general knowledge as in "It is always windy near tall buildings". Definite descriptions implicitly make use of quantification as the example "the people of China" shows. Lastly, the meanings of complex concepts require quantification. Any complex action concept like <u>walking</u> has associated with it as part of its definition assertions such as "at all times, some of the limbs of the individual engaged in walking support the individual (see section 4.4).

There seem to be three methods in conceptual dependency theory for expressing universal quantification. The first method involves the use of variables assumed to be universally quantified, as in "if <u>one</u> smokes this may cause <u>one</u> to get cancer." Here "one" stands for any person. It is not clear whether a similar approach is envisaged for universal quantification over other (nonhuman) sets. In any case this

device is inadequate, as it does not allow for multiple quantification, e.g. "Any politician can fool some of the people all of the time".

A second method for expressing universal quantification is the inference rule. We might have an inference rule in our system that determines "If X is thirsty, infer that X will drink something" where X is a universally quantified object that stands for any person. Thus a machine might easily answer a question like "John is thirsty. Will John drink something?"; however, the existence of this inference rule will not allow a machine to answer questions like "Will John drink something _if_ he is thirsty?" and "Do thirsty people drink liquids?", since no assertion to the effect that someone is in fact thirsty has been made. The problem is that we lack accessibility to a procedurally encoded piece of knowledge _as a fact_. In other words, knowing how to use a fact does not guarantee knowledge of the fact.

The use of the conceptual tenses _timeless_ and _continuing_ is yet another method for expressing quantification. Schank appears to use the timeless tense to designate _habitual actions_, for example, "John sells cars". The continuing tense is closely related. It is used in the sense of _activity_ as defined in Evans (1967). However, these special devices do not address the general problem of quantification.

Quantifiers are expressed in preference semantics by

relating them to a primitive semantic unit of the form _type indicator_, or, like Schank, in what Wilks called common sense inference rules. Relating quantifiers to a semantic unit of the form type indicator is used purely as a device for translation.

Basically the problem in semantic network representations of quantification is that of indicating the _scopes_ of universal and existential quantifiers, which presents little difficulty in predicate calculus representations. The notation used in this formalism is analogous to quantifier-free normal form in predicate calculus. Propositions are in _prenex form_ (i.e., quantifiers have maximum scope), existentially quantified variables are Skolemized, and universal quantification is implicit. This first of all requires a distinction between existentially and universally quantified nodes. A simple method is the use of _solid lines_ for existentially quantified concept nodes (as in all previous figures), and _broken lines_ for universally quantified nodes. Graphical Skolemization then consists of linking each existentially quantified node to all universally quantified nodes on which it depends (i.e., whose universal quantifiers precede the existential quantifier in prenex form). _Dotted lines_ serve as these dependency links for easy distinguishability from propositional and logical links. For example, "All dogs chase some cat" is represented as shown in Figure 3.7.

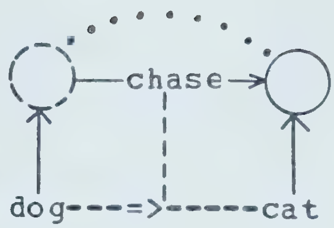


Fig. 3.7 "All dogs chase some cat"

In predicate calculus notation this is

$$(Ax)\{dog(x) \Rightarrow (Ey)[cat(y)\&chase(x,y)]\}$$

or $dog(x) \Rightarrow [cat(f(x))\&chase(x,f(x))]$, Skolemized. Now if we can assume $(Ey)cat(y)$, i.e., there is at least one cat (or alternatively, that there is at least one dog), then this becomes

$$cat(f(x)) \;\&\; [dog(x) \Rightarrow chase(x,f(x))],$$

which corresponds to the slightly simpler diagram shown in Figure 3.8.

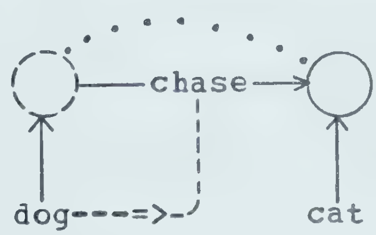


Fig. 3.8 "All dogs chase some cat"

Here the "cat" proposition is no longer regarded as a consequent of the "dog" proposition. This type of simplification is often appropriate for encoding natural language statements, since we do not usually communicate in terms of propositions which are trivially true by virtue of the nonexistence of their referents (which is not to say that we do not communicate about nonexistent entities). The diagram

for the proposition "There is a cat which all dogs chase" differs from Figure 3.8 only in the absence of the dependency link between the "cat" and "dog" nodes.

Time calls for special treatment because of its central importance in structuring events. Pairs of parentheses are used instead of circles for moments of time and pointers to moments of time are marked "T". A name for a moment of time can be placed between the parentheses. Intervals of time are represented as pairs of square brackets. Broken brackets and broken parentheses indicate universally quantified time variables. Pointers to time nodes may be suppressed altogether by placing times directly alongside the predicate tokens to which they apply. With this notation, complex time relations can be represented quite conveniently. The sentence "there is always someone there" is diagrammed in Figure 3.9. Additional conventions for time will be introduced in section 3.2.6.

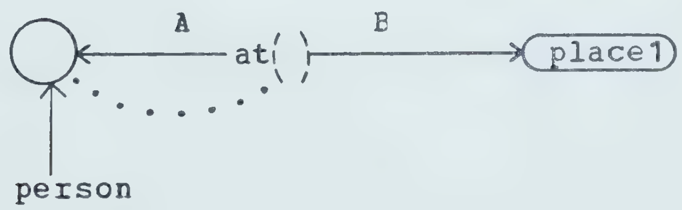


Fig. 3.9. "There is always someone there"

The proposed method of representing quantification is applicable only to propositions in prenex form. No generality is lost (though clarity is sometimes compromised) as a result of this restriction, as long as we are dealing with a purely

extensional logic (roughly, one in which all propositional connectives are truth-functional). However, propositions involving (nonextensional) modal operators such as "necessarily" and "believes" cannot be converted to prenex form. To facilitate the representation of such propositions, a generalization of the present notation is introduced in section 3.2.7 which allows arbitrary embedding of quantifiers.

Many higher-order constructions are easily expressed with the notation already introduced. For example, "John has all of his father's faults, and carelessness is one of them" is represented as shown in Figure 3.10.

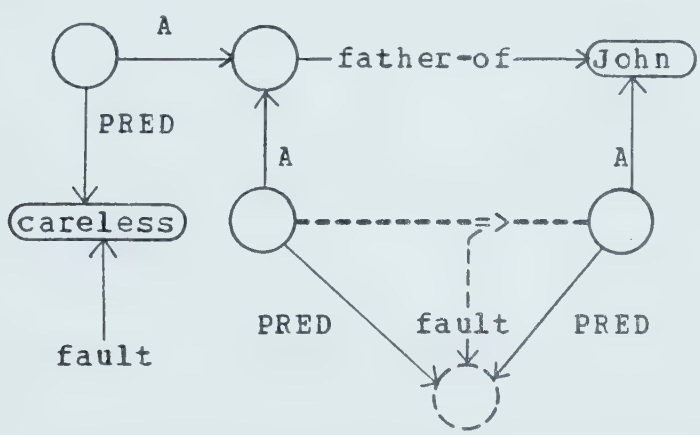


Fig. 3.10 "John has all of his father's faults, and carelessness is one of them"

Note that both the abbreviated and unabbreviated notation for propositions have been used here. The higher-order predicate is "fault", and the universally quantified node should be read "for all predicates". Alternatively we might define the set of faults of John's father and predicate each member of the set about John. "Careless" would be asserted to be a member of this set. Figure 3.11 illustrates this alternative

formulation. Let "$\in$" denote set membership.

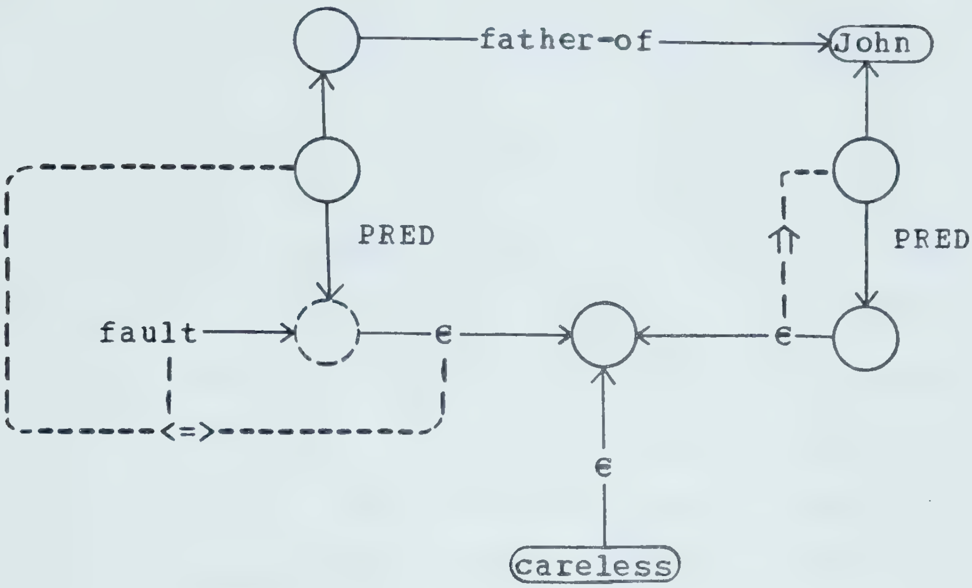


Fig. 3.11 "John has all of his father's faults, and carelessness is one of them"

Conventions for abbreviating implication and equivalence assertions of the type appearing in Figure 3.11 will be mentioned shortly.

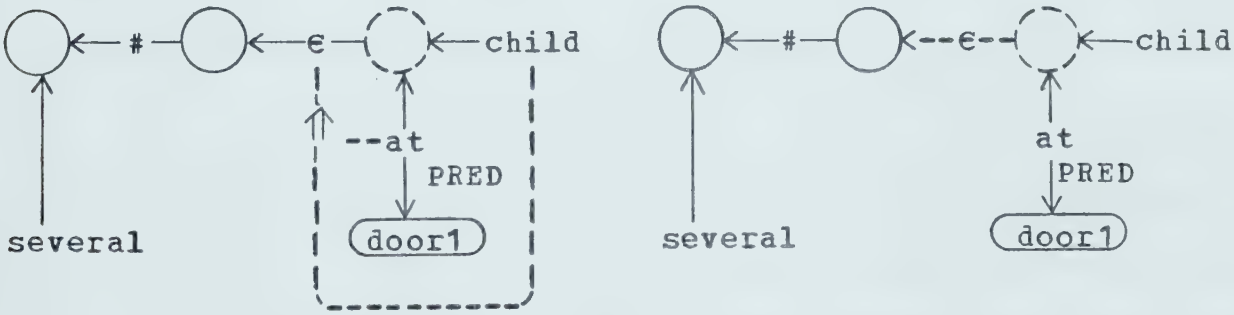
Past claims about the equivalence of certain varieties of semantic network notation to second (or higher) order logic have not been backed by adequate quantificational apparatus. Statements about predicates alone do not demonstrate a second-order capability, as they can be made in a many-sorted first-order logic. Logical quantifiers remain unsuitable for expressing many natural language quantifiers. Natural language quantifiers not readily expressible in terms of the logical quantifiers, such as "several", "many", "most of", "a few more than", etc., can be handled systematically by the use of

(fuzzy) properties of set cardinality and relations between set cardinalities, plus standard set relations such as set inclusion.

We can classify natural language quantifiers along a spectrum with absolute indicators of set size at one end, and comparative indicators of set size at the other, i.e. those comparing the size of one set to that of another set. The logical quantifier E (there exists) belongs to the first of these categories, since (Ex)P(x) tells us that the set of P's contains at least one member. The quantifier A (for all), by virtue of its equivalence to ¬E¬, can also be placed in the first category. In the context (Ax)(P(x)=>Q(x)) however, where the number of P's is finite, it can alternatively be placed in the comparative category. It tells us that the subsets of P's that are Q's is as large as the set of P's itself. Common absolute quantifiers are "none", "one", "two", "three", ... , "several"; common comparative quantifiers are "all of", "most of", "a small fraction of", "a slight majority of", "one-half of", "two-thirds of", "as many as", "twice as many as", etc. Some quantifiers show both absolute and comparative attributes, especially "some" and "many". For example, in "Many artificial satellites are orbiting the globe" "many" is used absolutely - it appears to imply a cardinality of at least about a dozen. In "Many students attend John's class" "many" is used in the sense "considerably more than attend the average class". This particular use of "many" is discussed

quite satisfactorily in Bartsch and Vennemann (1972). They do not appear to be aware of the absolute indicativeness of "many", however, nor of its comparative use in selecting a <u>subset</u> of another set, as in "Many of the world's people are undernourished". Contrast the numerical indication here with that in "many of the apples in the basket were rotten".

In any case, recognising the absolute/comparative behaviour of quantifiers, we can characterize them systematically by means of predicates on set cardinality and on pairs of set cardinalities as in Figures 3.12 and 3.13. In Figure 3.12 (b) the convention for abbreviating implication is shown, i.e. single broken lines being used for the conjoined antecedents and solid lines for the conjoined consequences. Let "#" denote set size (cardinality). Equivalence of two sets of propositions is abbreviated in an analogous manner in Figure 3.13, the members of one set being attached to a node with double broken lines and the members of the other set with solid lines.



(a) (b)

Fig. 3.12. "Several children were at the door."

We are regarding "#" as a function from sets onto integers and

"several" as a (fuzzy) property of numbers. If instead we regarded "several" itself as a possible value of set size, then it would not be possible to talk about <u>the</u> size of the set, as "#" would be many valued (eg., a 6-element set might have both size "6" and size "several"). In the next example "many" is expressed in two parts, the first being an <u>absolute</u> indicator of size (about a dozen or more?), and the second comparing set size to an average set size, as in Bartsch and Vennemann (1972). In the construction "avg #" is regarded as a function on classes of sets whose value is the average cardinality of the sets in a given class of sets.

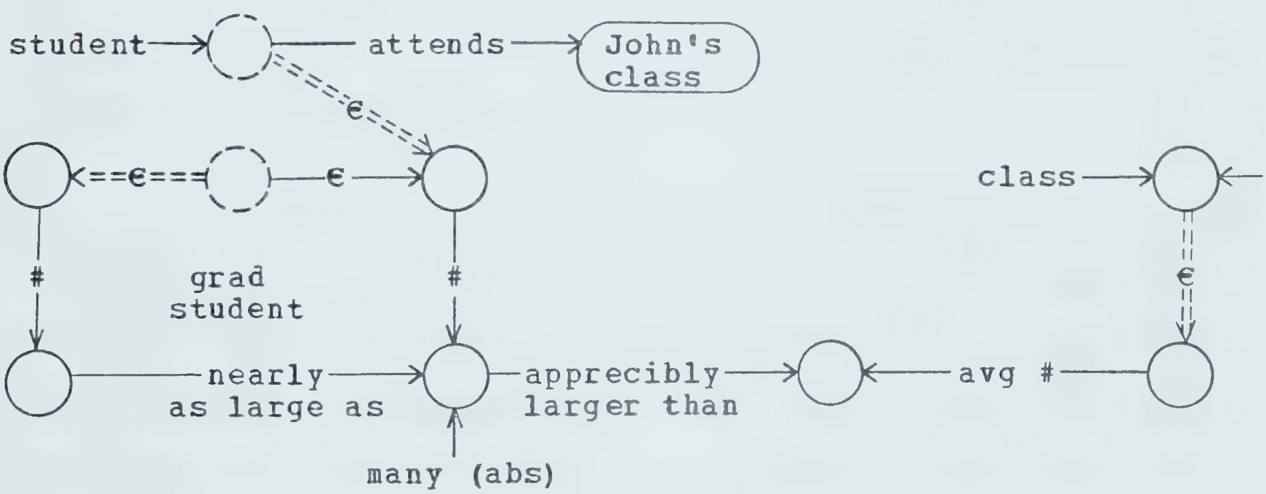


Fig. 3.13. "Many students attend John's class. Most of them are graduate students."

The systematic representation of logical and natural language quantifiers based on predicate calculus expressions as described in this Chapter makes all of the inference procedures of predicate calculus formalisms available to advocates of semantic networks.

3.2.4 Descriptions

The distinction between definite and indefinite descriptions has usually been ignored; therefore the method of representing both definite and indefinite descriptions is shown here. This section does not extend the expressive power of the network formalism. The representation of both definite descriptions ("the little old lady at the door") and indefinite descriptions ("a big apple") are based on the conventions for logical connectives and quantifiers already introduced rather than on a description operator (e.g. Moore, 1973). Description operators appear to be useful only at a superficial level of language representation, and in the domain of pure mathematics.

In Figure 3.14 the sentence "John's car is red" is diagrammed. A presupposition conveyed by the description "John's car" is that John has exactly one car (at least this is true for certain discourse contexts for which the given sentence might occur). One way of representing the sentence is shown in Figure 3.14(b), preceded by the non equivalent proposition that John owns a red car, and followed by an abbreviated version of (b). Note that the uniqueness condition has been expressed in (b) with the aid of equality.

The same convention also turns out to be useful for definite descriptions of <u>sets</u>, such as "the French-speaking people of Canada". This description is shown in full and

contracted notation in Figure 3.15.

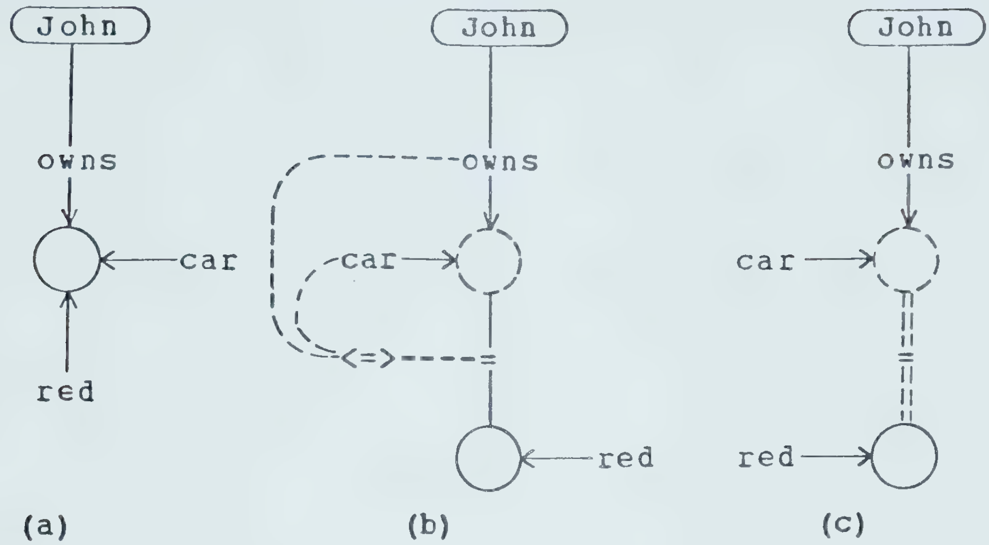


Fig. 3.14 (a) "John owns a red car";
(Ex)[owns(John,x) & car(x) & red(x)]
(b)&(c) "John's car is red".
(Ex)(Ay)[owns(John,y)&car(y)<=>x=y]&red(x)

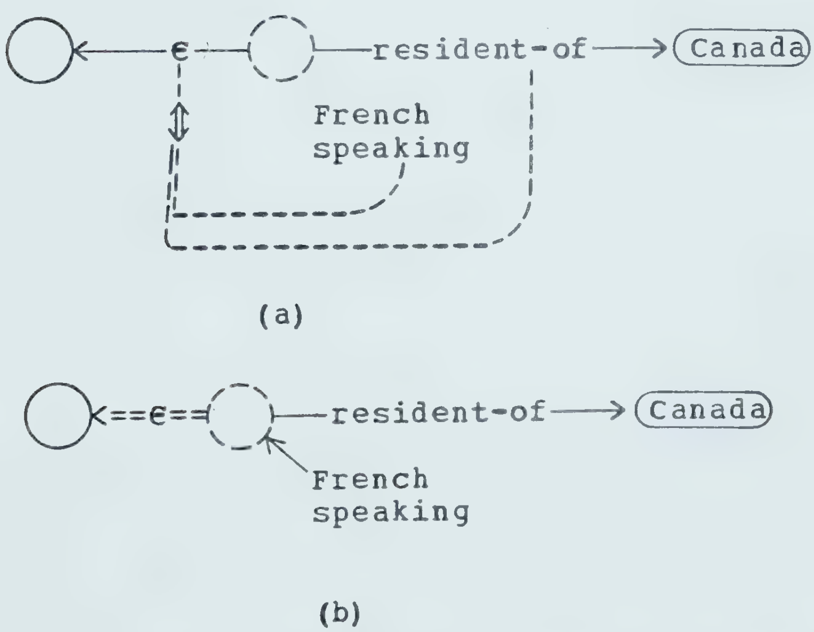
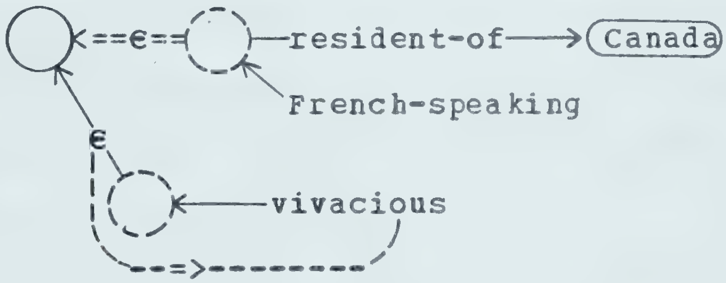


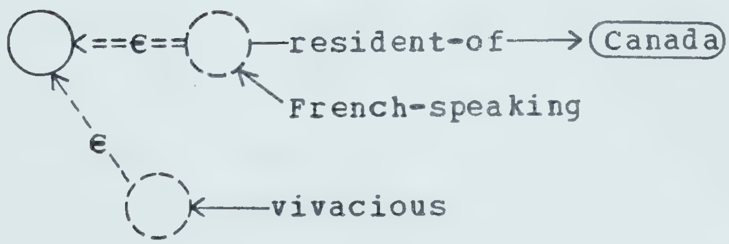
Fig. 3.15 "The French-speaking people of Canada"
(ES)(Ax)[member(x,S) <=>
French-speaking(x)&resident-of(x,Canada)]

A hazard in the contracted notation is that one might incorrectly read the operands of the implicit equivalence

incorrectly read the operands of the implicit equivalence operator as independently asserted propositions. In fact only the equivalence is asserted. In using the abbreviated notation, we must add another universally quantified node to make statements about the members of the defined set (any statements made about the node bearing the set definition will become part of the definition). Thus we could diagram "The French-speaking people of Canada are vivacious" as in Figure 3.16. But the exact analogue of the equivalence convention can be adopted for implication, single broken lines being used for the conjoined antecedents and solid lines for the conjoined consequents. Thus Figure 3.15 is redrawn as shown in Figure 3.16.



(a)



(b)

Fig. 3.16 "The French-speaking people of Canada are vivacious"

3.2.5 Lambda Abstraction

The notation so far introduced is inadequate for descriptions of predicative concepts expressed in terms of predicates of the same (rather than higher) type. For example, suppose we wish to say that the property "human" is the same as the property "rational animal"; note that the latter property is of type 1 and is expressed in terms of the type 1 properties "rational" and "animal". We cannot diagram this statement on the basis of the formula (Ax)[human(x) <=> rational(x) & animal(x)], since this merely asserts extensional identity (i.e., the set of human beings equals the set of rational animals). The desired statement of intensional identity can be made with the aid of Church's lambda operator. This operator abstracts a predicate from an open sentence by designating certain variables of the sentence as arguments of the predicate. Thus we write

human = lambda x [rational(x) & animal(x)].

A more interesting example is provided by the sentence "Loving one's neighbours is a virtue", which requires abstraction of the monadic predicate "loves one's neighbours" from the dyadic predicate "loves". This is shown in Figure 3.17, using a graphical analogue of lambda abstraction.

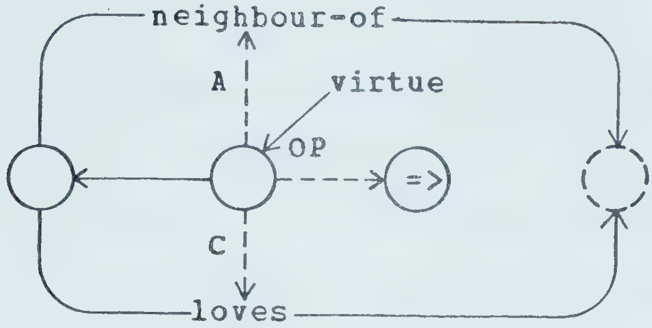


Fig. 3.17 "Loving one's neighbours is a virtue"
virtue{lambda x[(Ay)[neighbour(x,y)=>loves(x,y)]]}

Lambda conversion is accomplished by means of a (solid) lambda link from the proposition expressing that an individual "loves all of his neighbours" to the node for the individual. In general graphical lambda abstraction involves the construction of some open sentence (possibly with embedded lambda expressions), and lambda conversion of some of the variables of the sentence. Open sentences are formed exactly like propositions, except that some of the participating concept nodes are regarded as free variables. Lambda conversion is symbolized by lambda links from the node corresponding to an open sentence to free variables of the sentence. In the nonmonadic case lambda links are labeled lambda A, lambda B, etc., (or in some other systematic way) to distinguish the arguments of the abstracted predicate. Other examples are shown in Appendix D, Figures D.7 and D.8.

3.2.6 *Time*

Time is regarded as the only situational (Cf. McCarthy and Hayes, 1969) or contextual variable that needs to be added to action propositions. This is in contrast to Anderson and Bower (1973), Rumelhart et al. (1973), and Schank et al. (1973).[9] Locale is added to time in their representations as basic dimensions of events. But locale is not a property of events as a whole, but a (frequently time-dependent) property of the participants in an event. For example, in "John is watching a circling hawk" it is John and the hawk who have locations, not the event.

We start by allowing time to be used in two modes - *instantaneous* and *interval*. In the instantaneous mode a proposition can have either a fixed or variable moment of time associated with it. In the interval mode the "moment" is replaced by a time interval. The interval can be omitted in particular contexts (where it would normally be given) to simplify propositions that describe states and events with more enduring properties (like being a girl, car, etc.). This omission is a matter of expediency; any change involving a

---

[9] Whenever a proposition is regarded as true only within a particular situational context, that context can be made an explicit premise instead of an argument of the predicate. Thus "Mary is livelier with her lovers than with her parents" (Bartsch and Vennemann, 1972) would be rephrased as "For all times t and all times s, if Mary is with a lover at time t and if Mary is with her parents at time s, then Mary is livelier at time t than she is at time s".

metamorphosis (like a girl becoming a woman or a caterpillar becoming a butterfly) would require explicit recognition of time dependencies.

Within this framework, temporal relations including tenses (which can be built up from more elementary temporal relations) can be defined. If we restrict our view of time as consisting of a set of elements (time points) and a relation that partially orders them, we can define binary temporal relations similiar to those of Bruce (1972) or Schank et al (1973).[10]

Bruce, in particular, has given a systematic method for defining tenses by mapping various time relations given by auxiliary verbs and the form of the main verb and his seven binary ordering relations on time segments, e.g. "I had gone" - maps to "after". Thus a tense is an n-ary relation on time segments, e.g. past tense is one where the relation "after" holds between two time segments.

With our modification to some of the binary ordering relations, we can then represent a sentence like "While he was

---

[10] We would modify some of the binary ordering relations defined by Bruce to use endpoints of time segments (his definition) in the following way. Instead of defining after(X,Y) iff $x2<y1$ where X, Y are time segments with endpoints $(x1,x2)$ and $(y1,y2)$ respectively, we would choose to define after(X,Y) as after(X,Y) iff $(Ax \in X)(Ay \in Y)[x \geq y]$. This allows the event associated with time segment X to terminate at precisely the same time as the event associated with time segment Y initiates.

in Rome, before he met his murderer, he first sang in La Traviata" as in Figure 3.18.

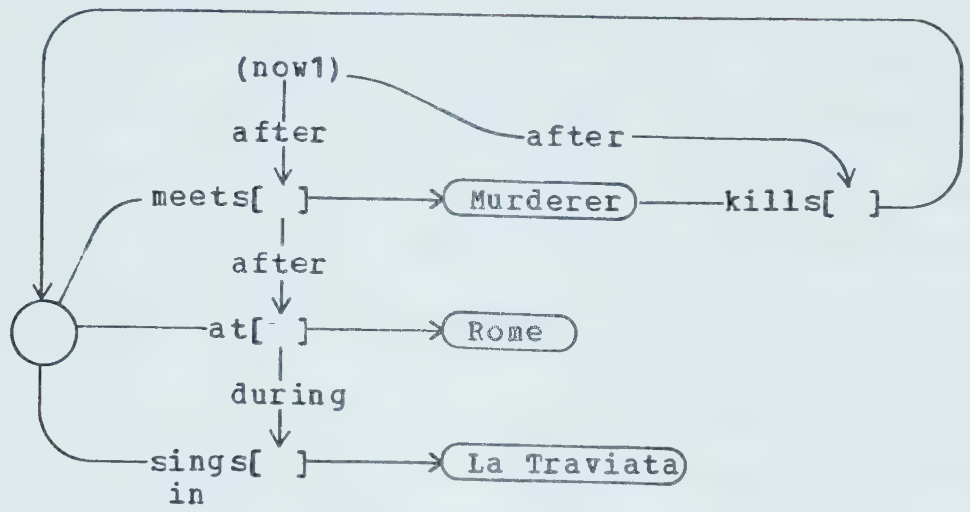


Fig. 3.18 "While he was in Rome, before he met his murderer, he first sang in La Traviata."

3.2.7 Modalities

Additional conventions are required to represent propositions involving modal operators such as the necessity operator, the belief operator, the causal operator, and the counterfactual conditional.[11]

The only notational problem raised by the introduction of modal operators appears to be that of distinguishing between "opaque" and "transparent" environments generated by

---

[11] Note that it is only the representation, not the manipulation or formal semantics of modal constructions that is at issue here. However, it is reassuring that the "possible worlds" semantics devised in recent years by modal logicians appears to provide an adequate basis for the formal semantical analysis of modal constructions.

such operators. A sentential environment is opaque if replacement of a term by a referentially equivalent term can change the truth value of the sentence, and transparent otherwise. For example, the necessity operator generates an opaque environment in the sentence "9 is necessarily greater than 8", as we cannot replace "9" by the referentially equivalent term "the number of major planets".

In general we can regard a term in an opaque environment as locked into the <u>scope</u> of the modal operator (necessarily, wants, etc.). Since quantifiers in such terms cannot be extracted to convert the proposition to prenex form, the earlier notation for quantifiers is inadequate. We need constructions that allow arbitrary embedding of quantifiers within the scopes of propositional connectives (extensional or otherwise). To see how the prenex restriction can be removed consider the following proposition: (Ax)(Ey)(Az){(As)P(s,x) => (Et)[Q(t,y,z) <=> (Au)(Av)(Ew)R(u,v,w)]}. We can completely specify the scopes of all quantifiers as follows. First, for each sequence of <u>adjacent</u> quantifiers, we specify the dependence of the existentially on the universally quantified variables as before, i.e., y depends on x and w depends on u and v. Second, we specify which variables have their quantifier scopes nested just inside the scopes of which propositional connectives. Thus s and t depend on =>, and u, v, and w depend on <=> in this sense. Exactly the same kind of dotted dependency link can be used for this second type of

scope relationship as for the first. The network representation of the above proposition is shown in Figure 3.19.

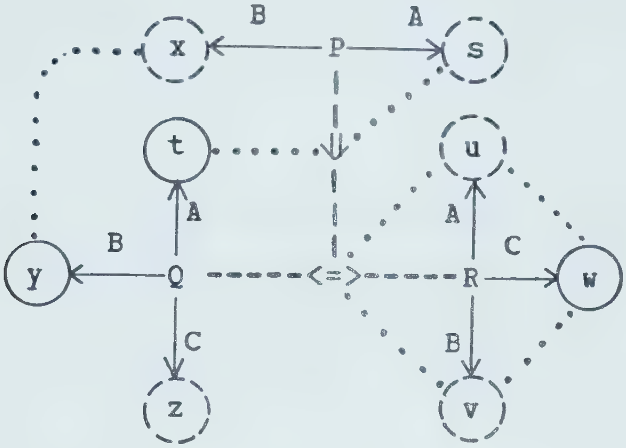


Fig. 3.19 Indicating quantifier scopes.
(Ax)(Ey)(Az){(As) P(s,x) =>
(Et)[Q(t,y,z) <=> (Au)(Av)(Ew) R(u,v,w)]}

By the transitivity of scope inclusion (Au), (Av), and (Ew) lie within the scope of =>, since the equivalence proposition is embedded within the implicative proposition. The assumption that all quantifiers have <u>maximum</u> scope compatible with the indicated constraints uniquely determines all scopes.

We now apply this notation to the representation of opaque constructions, beginning with the earlier example involving necessity. The representations of the nonsynonymous sentences "9 is necessarily greater than 8" and "The number of major planets is necessarily greater than 8" are shown in Figure 3.20.

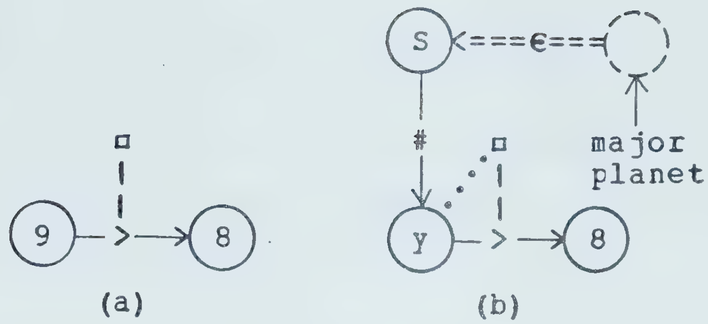


Fig. 3.20 (a) "9 is necessarily greater than 8"
□(9>8)
(b) "The number of major planets is necessarily greater than 8"
(ES)(Ax){member(x,S)<=>major-planet(x) & □(Ey)[#(S,y) & y>8]}

Counterfactual implication can be treated in much the same manner as necessity. In the sentence "If there were a major planet beyond Pluto, the number of major planets would equal 10" the term "the number cf major planets" is nonreferential. The representation is shown in Figure 3.21.

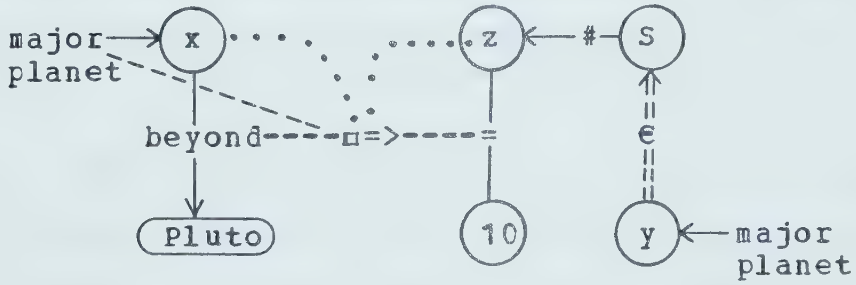


Fig. 3.21 "If there were a major planet beyond Pluto, the number of major planets would equal 10"
(ES)(Ay){[member(y,S) <=> major-planet(y)]&[(Ex) major-planet(x)&beyond(x,Pluto)]□=>(Ez)[#(S,z)&z=10]}

We have borrowed Lewis' symbol □=> to symbolize counterfactual implication (Lewis, 1973); however, any number of implicitly conjoined antecedents and consequents are allowed as operands, much as in the generalized form of material implication.

Many English modal sentences, particularly those involving verbs of propositional attitude such as "wants" or "believes", admit both a transparent and an opaque reading. An example is given in section 4.7.

Propositional attitudes may involve quantification over propositions, as in "John knows everything", or in "John knows everything Mary knows". The most direct way of representing such propositions is by quantification over propositional variables, as in Figure 3.22.

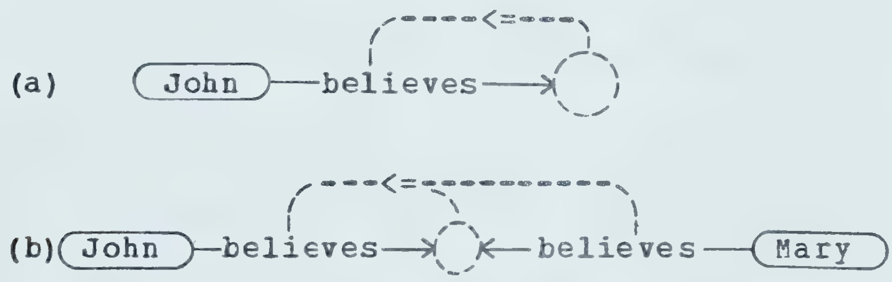


Fig. 3.22 (a) "John knows everything"
(b) "John knows everything Mary knows"

In Figure 3.22(a) "John knows everything" has been paraphrased as "John believes all true propositions". If we accept these representations, we must carefully distinguish between the universally quantified "proposition nodes" in Figure 3.22 and the propcsition nodes previously introduced as points of attachment for the parts of an explicit proposition. These previous proposition nodes are not quantifiable. The propositions in which they participate "exist" simply by virtue of appearing in the semantic net. The universally quantified nodes in Figure 3.22, by contrast, are <u>concept</u>

nodes dencting <u>complete</u> propositions.

We could avoid the use of propositional variables altogether by paraphrasing "for all propositions" as "for all predicates P and all x such that P(x)", giving the new version of Figure 3.22(a) shown in Figure 3.23.

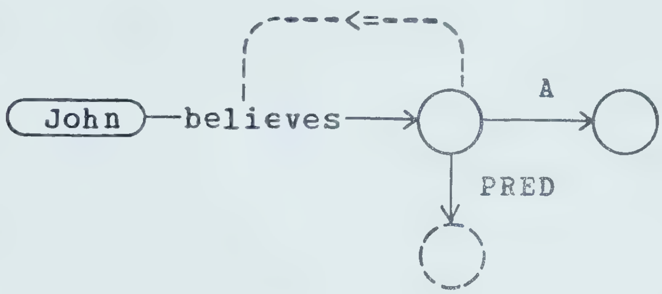


Fig. 3.23 "John knows everything"

This would avoid the use of propositions as concepts. On the other hand Figure 3.22 has the advantage of being more concise. Note that the replacement of quantification over propositions by quantification over predicates and variables can also be carried through for existential quantification. For example, the sentence "There is a proposition x such that if x is true then mankind is doomed" can be paraphrased as "There is a predicate P such that if P(x) is true for all x then mankind is doomed".

The techniques for representing knowing and believing apply equally to other propositional attitudes such as remembering, supposing, intending, deciding, avoiding, hoping, imagining, pretending, and trying. Nonreferential terms within the scopes of such operators (whichever ones are deemed useful

independently of the others) can be identified by means of scope dependency links as in Figures 3.20-3.23. The same applies to the deontic modalities such as obligation. It should be obvious, for example, how "John ought to marry the prettiest girl" would be represented.

As a final important modality, causal dependency should be mentioned. Causal explanations are rather closely related to counterfactual conditionals, as can be seen from the inference "B would not have happened if A had not happened", which is often reasonable given that "A caused B". An example of an opaque context generated by a causal construction is found in the sentence "John asked Mary to dance because she was the only girl left without a partner". Substitution of the term "Mary" for its referential synonym "the only girl left without a partner" clearly fails. In causal constructions as in other modal constructions, therefore, we may need scope dependency links.

3.3 Discussion

In extending semantic network notation to include constructions like logical connectives, quantifiers, descriptions, and modalities, the question of whether we lose the visual suggestiveness of other less inclusive semantic network representations arises.

The preceding is intended to establish a notational

basis for state based theory and is particularly well suited for that purpose. Certainly a rather different representation might be adopted to meet specific needs in specific applications of semantic networks. Also a different or more extensive set of propositional operators might be formulated. For example, the following operator ("subconcept") may well be useful in representing the essential meaning relationships among concepts: an n-ary relation P(x1,...,xn) is a <u>subconcept</u> of an n-ary relation Q(x1,...,xn) if $\Box$(Ax1)...(Axn)[P(x1,...,xn) => Q(x1,...,xn)]. Thus "crow" is a subconcept of "bird" (i.e., crows are necessarily birds), "drinks" is a subconcept of "ingests" (i.e., if x drinks y then x necessarily ingests y), etc.[12] The point is that this operator and other compound operators can be expressed in terms of the basic network notation.

A variety of issues in the representation of informal knowledge could raise additional notational problems. Examples are the handling of vagueness, events, the lexical meanings of complex concepts, and overall knowledge organization. Beyond these relatively static issues lie the more dynamic issues of actual language interpretation and generation, plausible inference, learning, and the interplay between procedural and

---

[12] This differs from Quillian's "subset" relation in two respects. First, "property P" is not regarded as synonymous with "the set of P's". Second, Quillian presumably had in mind the usual (contingent) subset relation, rather than one reinforced by necessity.

factual knowledge. Clearly any questions about representation raised by these problem areas can only be answered in the context of particular approaches towards the solutions of the problems themselves. One such approach is discussed in Chapters 4-6.

Chapter 4

STATES, COMPARATIVES, AND ADVERBS

A general theory of natural language understanding would require a representational schema with sufficient expressive power to represent the meaning content of ordinary language. The extended semantic network notation presented in Chapter 3 possesses this expressive power. The discussion in this Chapter is a development of some ideas concerning the representation of individual items of "factual knowledge" in a computer, where this knowledge is thought of as being conveyed to the computer in natural language. Parts of this chapter are taken verbatim from an earlier report (Cercone and Schubert, 1974).

4.1 The Basic Framework

The basic framework embodied by many natural language understanding systems is the <actor-action-object> formalism. Certainly this is not without justification. Much of natural thought and communication follows from this framework. Only a deeper analysis of actions and intentions seems to belie thinking of this framework as underlying natural language. While not denying the intrinsic value to organisation, heuristic programming, and pragmatics that this <actor-action-object> formalism suggests, its theoretic value for beginning

investigations into language comprehension is minimal.

A more fundamental starting point was chosen for this research using extended semantic networks to represent many natural language constructions. Explicit comparisons are made below between Schank's (1972) conceptual dependency theory and Wilks' (1973) preference semantics theory on the one hand and the extended network approach on the other. Conceptual dependency and preference semantics have given efforts into natural language understanding renewed impetus and have strongly influenced the development of this thesis.

Specific difficulties with conceptual dependency and preference semantics are considered in the following sections. Plausible ways of overcoming these difficulties are given using extended semantic network constructions. Perhaps the most important difficulties are those encountered when attempting to represent complex concepts and adverbials. However, other points need to be discussed first.

4.2 States, Events, Actions, Cases, Causes, and Intentions

Conceptual dependency diagrams and preference semantics templates are capable of expressing four sorts of assertions:

(1) state attribution - ascribing a modifier to an object or set of objects at some time;

(2) events - ascribing a change of state to an object or set of objects at some time;

(3) actions; and (4) causes.

However, a sentence such as

> The sun was turning red and approaching the western horizon. <4.1>

raises many questions about Schank's and Wilks' formalisation of these distinctions. In <4.1> the motion of the sun *must* be done *by* somebody or something whereas its change of colour *cannot* be done by somebody or something. Thus, using the <actor-action-object> formalism espoused by both Schank and Wilks, modes of behaviour which are expressed by actions *must* have actors whereas all other modes of behaviour *cannot* have actors. In the case of the (apparently) moving sun in sentence <4.1>, one is hard pressed to identify the actor; similarly in the sentence "The breaker was moving toward shore." Consequently we are compelled to regard certain ongoing activities which intuitively just "happen" as instigated by someone or something (including natural forces in a vague, unspecified sense).

Just as we are compelled to regard certain ongoing activities as instigated by somebody or something, we are denied the option of regarding certain actions as having an agent as shown in

> John was hurting Mary by pulling her hair. <4.2>

In <4.2> the "hurting" not being an action, has no actor whereas in

> John was dragging Mary by pulling her hair. <4.3>

the "dragging", insofar as it involves PTRANS'ing *does* have John as an actor.

We may wonder by what criterion we draw the line between what an actor <u>does</u> and what he <u>causes</u>. In <4.2>, according to Schank, we are to regard the "hurting" as <u>caused by</u> the "pulling" action. But the same is true of PTRANS'ing in <4.3>. Furthermore, even direct bodily action such as moving an arm can be viewed as caused by muscle contraction or, subjectively, as caused by an act of will, either of which again may have antecedent causes.

In view of these shortcomings of Schank's formalism, it seems to us that no structural primitives should be associated with actors at all. Instead we will propose a neutral representation in which events are expressed as sequences of states of the participants. The successive states simply express "what happened", without explicit commitment as to "who did it". However, the agent(s) in an event can be identified by supplementary propositions. Thus the notion of an agent can continue to be used to aid interpretation and inference. However, it would be regarded as a rather "fuzzy" higher level concept, understood by the system in terms of the role of a supposed agent within a sequence of causally and teleologically related states. For example, in the sentence "John uprooted the sapling" the term agent would be considered highly applicable to John's role in the event while in the sentence "The avalanche uprooted the tree" its applicability to the role of the avalanche would be considered relatively low. The notion of an "agent" seems to depend in part on

causal priority of a state of the supposed agent in the sequence of states under consideration, and in part on the extent to which purposive behaviour can be ascribed to the supposed agent in general, and in part to the extent to which the particular sequence of states which he initiated can be assumed to be intentional on his part.

Similarly we propose to separate <u>why</u> something happened (causes, enabling conditions, reasons, explanations, justifications, and the like) from <u>what</u> happened. As with "agents", this does not prevent us from including causal propositions in the representation and relying heavily on them for interpretation and inference. However, time relations and changes of state, not causes, will give coherence to a set of propositions as an "event".

We feel that Schank's <u>instrumental</u> relation between actions can and should be represented in terms of causation and intention.[12] For example, if a system has a conceptualization to the effect that John was PTRANS'ing the ball <u>by</u> PROPEL'ing it, then this conceptualization should also express that the PROPEL'ing was <u>causing</u> the PTRANS'ing. In fact, phrases ostensibly expressing instrumental actions often express no more than causation. An example is the "by" clause

---

[12] Schank uses instrumental case relations to attach supplemental conceptualizations to the main conceptualization where the supplemental conceptualization serves as means toward the end expressed by the main conceptualization.

in

> The effluents were killing the fish by
> raising the temperature of the water. <4.4>

When there is a difference, it lies in the intimation of purposive causation. In

> John woke Mary by blowing his trumpet. <4.5>

purposive causation is expressed, while in

> Mary woke up because John was blowing
> his trumpet. <4.6>

it is not. Sentences <4.5> and <4.6> clearly show that the instrumental relation amounts to a causal relation supplemented by intentional states.

According to Fodor (1972), actions are to be thought of as a proper subclass of events. Let us determine whether this is the case for Schank's notion of an action, and what might justify the special status of actions as opposed to events. According to Schank, an action is something a nominal can be said to be doing at some moment (this is not a quote, but an interpretation of Schank's definition). A study of his proposed inferences shows that in itself an action does not express a definite change in a situation; rather it expresses existence of a situation which tends to produce change, and all actual changes must be inferred. Formulas for actions in Wilks' theory are analogous although they are not described as explicitly as Schank's primitive action concepts. Actions, then, express modes of behaviour which promote but do not guarantee the occurrence of events. For example, the actions

PTRANS, INGEST, MOVE do not express changes in location; instead those changes are primary inferences given that an actor is PTRANS'ing, INGEST'ing, or MOVE'ing something. Syntactically, the relationship between an event, say a change in location, and the action, say PTRANS, whose primary inference is that event, corresponds quite closely to the relationship between verbs and their participles respectively. For example to say that John was PTRANS'ing himself with the result that his location changed is quite analogous to saying that he was going somewhere with the result that he went there.[13] In any case the term "action" is now seen to be quite misleading, since it normally connotes the occurrence of definite events, rather than the existence of a "dynamic" situation which tends to generate events.

Thus Schank's actions (contrary to the connotation of the term) correspond more closely to states than to events! To say that A is PTRANS'ing B is merely to express a momentary truth about the system in which A and B participate, not a change in that system (which remains to be inferred). This view is compatible with the observation that many common modifiers express subtle blends of "passive" and "dynamic" attributes. The examples below bring to mind conceptual images

---

[13] Unlike Schank we do not regard "he was going" and "he went" as equivalent; we claim that "he went there," unlike "he was going there," affirms that he did arrive at his destination, and that it is decidedly odd to say "he went there but didn't get there."

that illustrate a gradually increasing emphasis on dynamics.

blue sky
bright sun
glowing (or luminous) candle
burning candle
blazing fire
billowing smoke

Schank's actions, and, as far as we can determine, Wilks', are "dynamic states", or "activities", or "modes of behaviour" which _mediate_ changes in certain attributes. Thus PTRANS and MOVE _mediate_ changes in location, INGEST and EXPEL _mediate_ changes in containment relationships, and MTRANS _mediates_ changes in awareness.

We believe that the recognition that "actions" in Schank's sense are essentially states rather than events is important, since it leads to a uniform view of all (true) events as sequences of states. In this view the need for identifying "actors" of events does not arise, nor is it necessary to delineate the spurious boundary between "passive" and "dynamic" states.

We now illustrate our representation of states and events. Nothing new needs to be added to the network notation of Chapter 3. We regard any condition which can hold momentarily (blue, moving, running, etc.) as _a state_. Accordingly, any atomic proposition which is based on a time-dependent predicate is a _state proposition_. Figure 4.1 shows two concurrent state propositions: something (the redness of the sun) was _increasing_ throughout some time interval and

something else (the distance between the sun and the horizon) was <u>decreasing</u> throughout the same time interval.

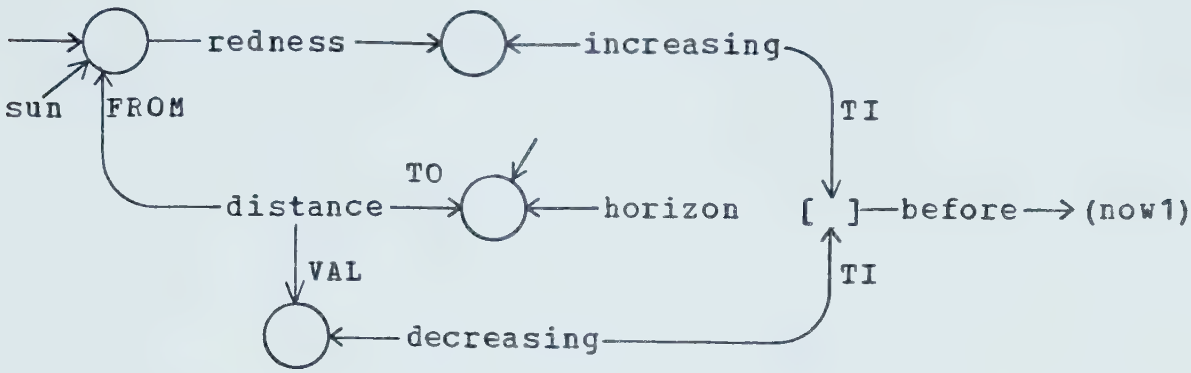


Fig. 4.1. "The sun was getting redder and approaching the horizon."

Actually there are two additional state propositions, concerned with the existence of unique values of redness and distance at all moments of time within the time interval of interest; these have not been made explicit since they can be taken to be implicit in the redness and distance relations.

<u>Events</u> involve a <u>change</u> in state as "the last leaf fell from the tree" illustrates. The definitive characteristic of state <u>changes</u> is the following: if a system has property A at time t1, and property B at time t2, then A-->B is a change of state if and only if A and B are mutually exclusive properties, e.g. A=solid, B=liquid; A=round, B=rectangular. In fact a state attribute such as colour which can assume various values can consistently be defined as a set of mutually exclusive properties, each member of the set being regarded as a value of the attribute. This admits both qualitative attributes such as colour as well as quantitative attributes

such as location. Figure 4.2 shows a simple event involving a single change of state

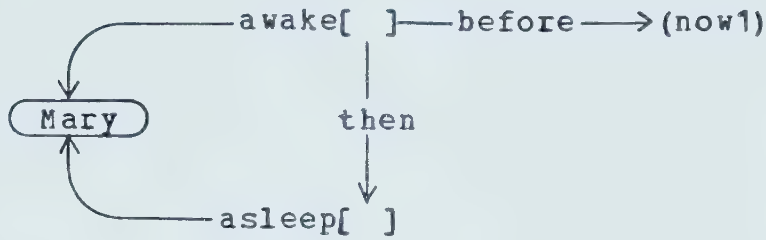


Fig. 4.2. "Mary fell asleep"

of a "system" with one component (Mary). The time relation "then" implies immediate succession of the two time intervals. Our representation cf one of Schank's standard sentences is shown in Figure 4.3.

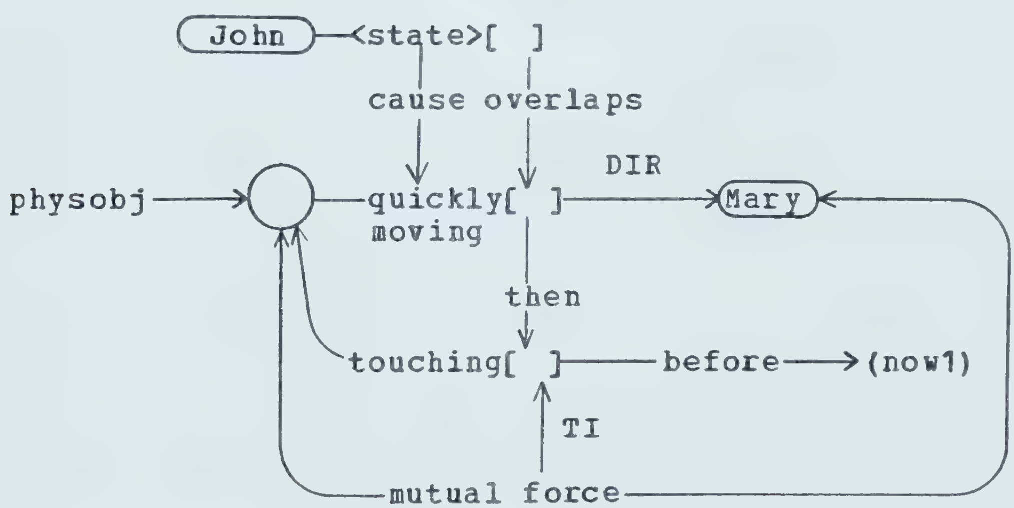


Fig. 4.3 "John hit Mary"

An explanatory paraphrase is the following. "Some unknown mode of behaviour of John caused some object to move quickly toward Mary. Subsequently the object reached Mary and exerted a force on her." Note that we have a state and an event here, viz.

John's unknown state and the event of the object moving toward Mary and striking her. In accordance with our earlier remarks about causation, the causal connections between John's state and the ensuing event does <u>not</u> make John's state <u>part</u> of that event. Only exclusive and successive states of a particular system of objects form events. A natural inference in Figure 4.2 would be that John <u>intentionally</u> hit Mary, i.e. that the missing state of John is that he was <u>trying</u> to bring about the event in question. We would represent "trying" by the state predicate "x has active goal y at time t". An additional example is shown in Figure 4.4.

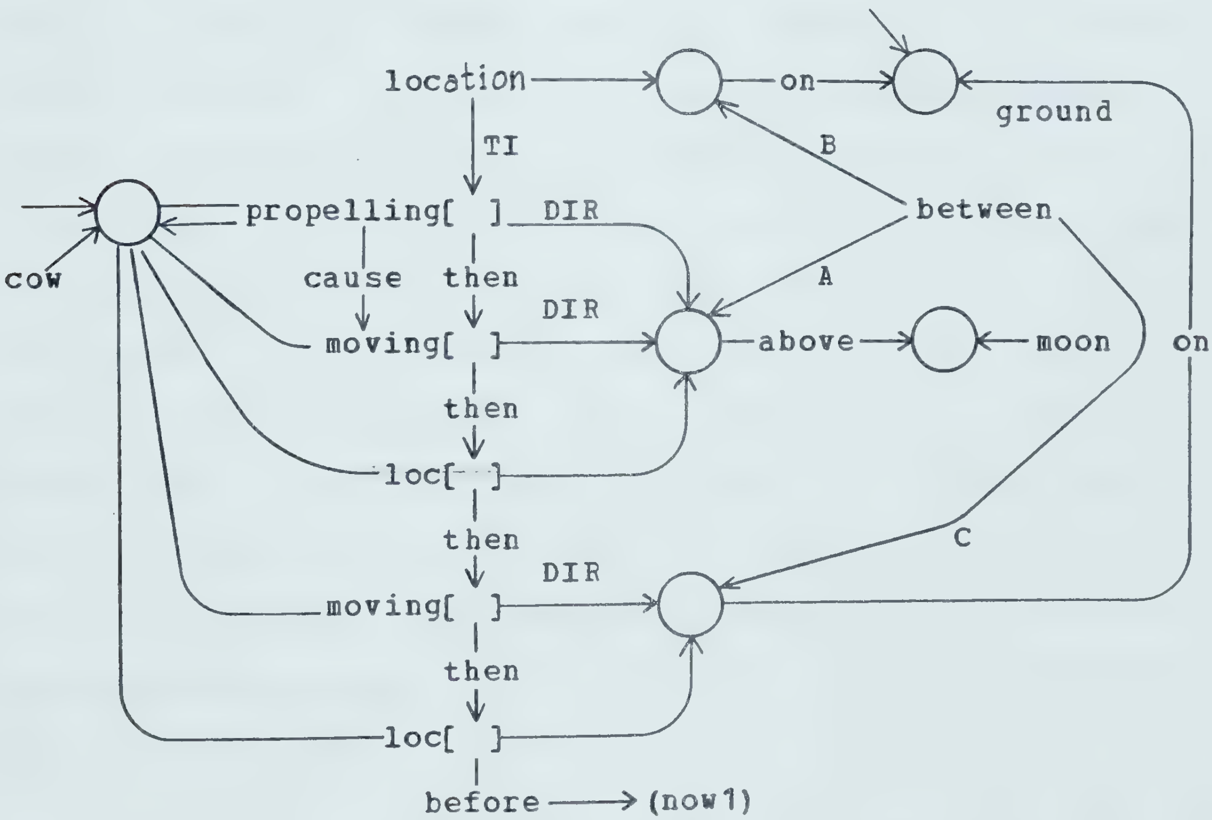


Fig. 4.4. "The cow jumped over the moon."

The explanatory paraphrase goes as follows. The cow was

somewhere on the ground, propelling itself towards a location above the moon; then it was moving toward that location; then it was at that location; then it was moving towards a place of destination on the ground, such that the moon is between the place of departure and place of destination; then it was at the place of destination. Note that "moving towards" could have been represented in terms of "distance decreasing" as in Figure 4.4.

An important consequence of our very broad conception of states is that new complex states (modes of behaviour) can be defined in terms of events involving primitive or already defined states. The time of occurrence of these events can extend some distance backward and forward from the moment at which the new state is defined to hold. For example "walking" is defined in terms of successive states of motion and displacement of the walker's feet and body over a "period of observation" encompassing (say) two steps, since an instantaneous "snapshot" of a person is insufficient for deciding whether or not that person is walking (although it may of course supply enough cues to prompt the inference that the person is walking). A tentative definition of "walking" will be given below.

Complex dynamic states (modes of behaviour) such as walking, running, dancing, tumbling, flickering, etc., can be constructed in terms of more elementary states. The constructions are necessarily as complex as the states they

describe. Complexity can result from the intricate coordination of several simultaneous activities (e.g. "rolling" expresses rotation and translation at coordinated rates), or from complex time dependencies (e.g. flickering), or from both (e.g. "walking" or even "building a snowman").

At this point the reader may wonder whether we propose to make any use of "cases" in our approach to representation. The answer is yes in the sense that we intend to exploit fully the semantic "preferences" that any given predicate induces on its arguments. For example the predicate "moving" prefers a physical object as its first argument and a physical location as its second argument; the predicate "has-active-goal" prefers a sentient being as its first argument and a state proposition as its second argument. Furthermore, there are broad similarities between the argument preferences of different predicates. For example, several predicates prefer animate objects in certain argument positions. We certainly can (and sometimes do) acknowledge such similarities and give a rough indication of the sort of preferences involved by using suggestive argument markers such as ANIM, THING, PLACE, DIRECTION, etc., instead of noncommittal markers such as A, B, C, ... , . However, we do not think that these markers can be chosen so that they express not merely similar but <u>identical</u> argument roles and semantic preferences, no matter in which predicate they occur. This view is supported by Bartsch and

Vennemann (1972):

> ... "case" is entirely a surface category and not, as Fillmore (1968) suggests, a category of universal semantics. Semantic representations are based on propositions, which consist of a relation (n-ary predicate with n>=0) with a finite number of arguments filled either with constants or with bound variables. The "meaning" of an argument as argument is entirely determined by its relation. Therefore, no two arguments have precisely the same meaning, as arguments. Thus, if the meaning of an argument as argument is called a case, then there are as many cases as there are arguments, and this number, if it is finite at all, is a very large one. What some linguists call "cases" are classes of arguments based on certain semantic similarities which follow from the semantic similarities of their relations. The fact that certain arguments show similarities in their syntactic behaviour, such as tending to occur in a certain position relative to the verb or belonging to the same surface case, does not support the assumption that there exists a small number of universal cases. Those syntactic similarities are simply a consequence of the fact that the human mind is structured in such a way that it tends to group objects on the basis of certain relevant similarities and then manipulate the objects of the group alike.

Thus semantic cases, while certainly useful heuristically in finding or inferring arguments of predicates have no universal or primitive status.

4.3 Complex Concepts

According to Schank's (1973a) dictionary, if X *walks* to Z (where X is human and Z is a location) then X PTRANS's X by X MOVEing the feet of X in the direction of Z. A minor criticism of this formula is that it rules out walking on one's hands and knees, or walking on one's hands (admittedly a rare skill). More importantly, the formula admits running,

skipping, hopping, jogging, shuffling, and even skating. Presumably, then, the dictionary entry is not intended to capture the full meaning of "walking" as we seem to understand it, but only those aspects which are most essential to language understanding and (immediate) inference.

Similarly Wilks' formulas are incomplete.[14] For example, it is correct to say that DRINK implies ((*ANI SUBJ)(((FLOW STUFF) OBJE)((*ANI IN)(((THIS (*ANI (THRU PART)) TO) (BE CAUSE)))) but not the converse (which could mean someone was receiving an enema). So again a selection of only some linguistically important features has apparently been made.

We feel that it is important to formulate more complete meaning representations for two reasons. First, we believe that somewhat more information will be required for adequate comprehension of "ordinary" discourse. Secondly, _much_ more information will surely be required to match the human ability to describe concepts and reason about them. For example, suppose we ask a reasonably articulate person to describe human "walking" _in as much detail as possible_. We might elicit at least the following information: Each foot of the walker repeatedly leaves the ground, moves freely in the walking direction for a distance comparable to the length of the walker's legs (while staying close to the ground), then is set down again, and remains in position on the ground, supporting

[14] Cf. Lakoff's (1972) lexical decomposition trees.

the walker, while the other foot goes through a similar motion. The repetition rate is about one repetition per second. The legs remain more or less extended. The body remains more or less erect and is carried forward at a fairly constant rate.

Further details could be added about flexing motions of feet, knees, and hips, the slight up-and-down motion of the body, typical arm motion, and forces exerted on the ground. Figure 4.5 shows a network which describes "walking" (regarded as a state predicate with three arguments besides time) along the above lines. A few propositions have been omitted so as not to clutter the diagram. These are that each foot is also above the ground (and close to it) while moving, that each foot is also supporting x while stationary; that the duration of each of the unlabeled time intervals [ ] is approximately half a second; and that the speed of motion of the walker's body is approximately constant. There is no difficulty in adding these state propositions, except that the last requires "moving" to have an additional argument, namely the speed of motion. Note that [ti] is the "time interval of observation" of the walker, and that it contains t, the time at which x is said to be walking. Thus "walking" is defined by behaviour in the temporal vicinity of the moment of predication, specifically about two seconds of motion allowing about three or four steps.

A limitation of our representation of "walking" is that

it is not applicable to unusual modes of walking (e.g. on hands and knees) or to animals. One question this raises is how many "kinds" of walking should be represented separately. Also, is there a representation which expresses the common features of all kinds of walking? We have attempted such a representation in Figure 4.6. The representation is based on the following characteristics of walking in general:

(1) it is done using limbs that are a subset of the limbs of the individual involved in the walking;
(2) the number of limbs involved is greater than or equal to two;
(3) at all times some of the limbs used for walking are in nonsliding contact with the walking surface (this is not the same as saying some of the limbs are in contact with the surface at all times);
(4) each limb used for walking is stationary on the walking surface at some time and subsequently is moving for some time; and
(5) the individual as a whole is in motion in the walking direction.

The interesting feature of our representation is the use of quantification to describe the role of any number of legs in the walking. Note that without quantification, describing the locomotion of say, a millipede would be very tiresome.

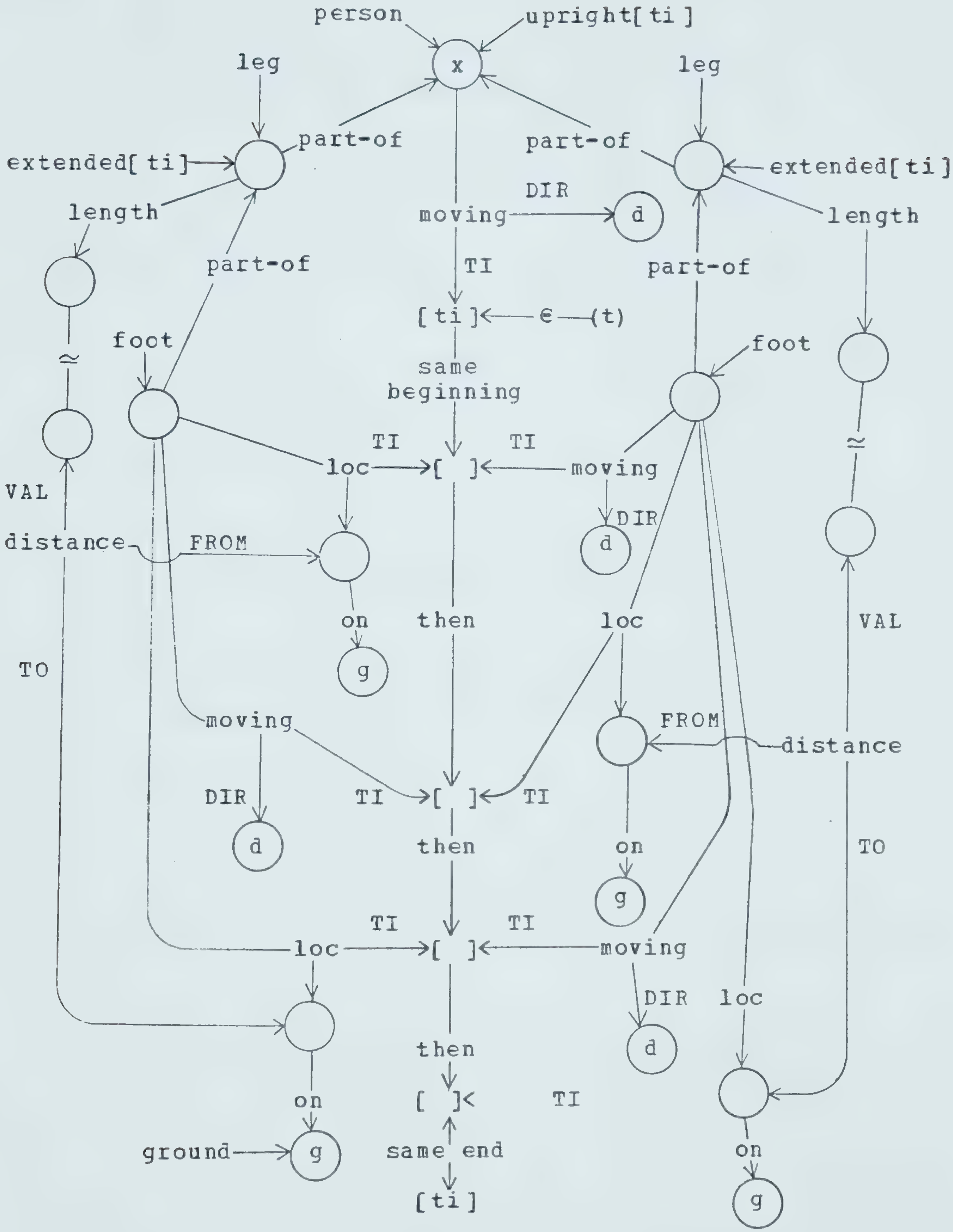


Fig. 4.5. "Person x walking at time t
in direction d cn ground g."

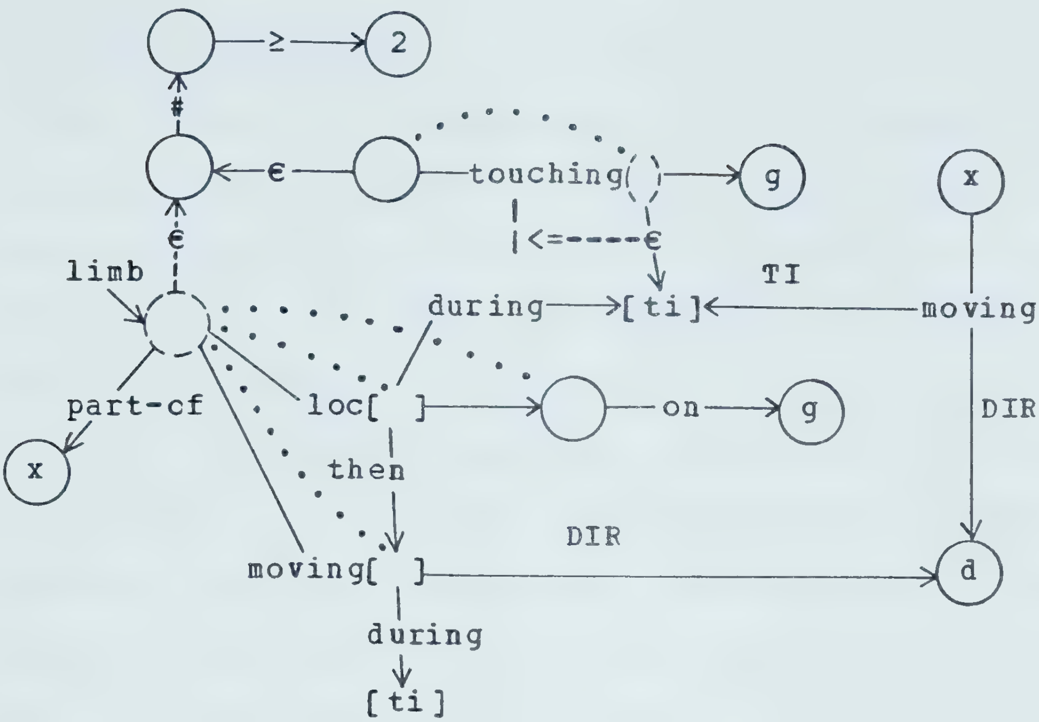


Fig. 4.6. "x walking at time t in direction d on surface g"

A serious flaw in our representation of "walking", and one for which we have no systematic remedy, is that we have ignored the "fuzziness" of many of the meaning components. For example, it seems necessary to put <u>some</u> constraints on the length of stride (lest the walker be allowed to mince forward in millimeter increments), yet to give an exact distance would be absurd.

An important consequence of conceptual fuzziness, with regard to the meanings of complex concepts, is that we can no longer draw a sharp boundary between extracting the <u>meaning</u> of an utterance and making probable <u>inferences</u> on the basis of the derived meaning structure. This is because we only find the <u>probable</u> meaning of an utterance. For example, the

utterance "John built the house" probably means that he built a large, rigid-walled enclosure with a roof, separate rooms, etc.; but none of this is certain. The utterance "John was laughing" probably means that he was producing a series of voiced sounds by staggered exhalation of air, and that his facial expression was merry; but he might have laughed silently, or his facial expression might have been derisive or even hostile. If we try to reduce semantic uncertainty by excluding from the "meaning" of a term all but its absolutely minimal content, and ascribe everything else to inference, we find that this doesn't work too well.[15] In the case of "house" all that would remain would be a "partial enclosure" - which accommodates a fenced-off field, a shipping crate, or a jacket. In the case of "laughing" we would perhaps be left with "spasmodic breathing and intent to convey amusement," which could suggest that John is asthmatic and dancing a jig.

Finally we wish to point out that many (perhaps most) concepts can be understood in different kinds of ways. For example, in "John was listening to the incessant chirping of

---

[15] It would seem that the human interpretative process does not proceed on the basis of the minimal conceptualization formed by embedding the minimal content of terms into the conceptualization. Rather the original term itself suggests what we could infer in addition to the minimal content. This seems like a reasonable view on the basis of efficiency considerations as well. It should be much simpler to insert probable inferences in a semantic structure by direct reference to the word definitions instead of analyzing the minimal representation and then looking for applicable inference rules.

the crickets," is "chirping" understood simply by its correspondence to a particular auditory sensation, or is it understood as a rapidly fluctuating, more or less uniformly high-pitched sound, or even as a complex variation of air pressure with time? Minsky's (1974) recent work on "frame systems" strongly suggests that the kind of understanding of a concept we use at a given time is extremely task - and context - dependent. This certainly casts doubt on the one-concept-one-formula approach to language understanding.

4.4 Adjectives and Relative Terms

A unified treatment of relative adjectives and comparatives is outlined in Bartsch and Vennemann (1972) which extends Montague's (1970) treatment within the intensional logic framework. Since their work is well documented and exemplifies the right kind of approach to the treatment of relative adjectives we see no need to explain it here. However, we level one criticism lest one unwarily falls victim to *prima facie* acceptance of their theory. The notion of a *reference set* (a set of objects whose members are used for comparison with some given object relative to some measurable attribute of the objects) is difficult to comprehend. For example, while it is possible to define a more or less adequate "reference set" to account for a phrase like "a large apple", it is not immediately apparent what the reference set would be if one were to ask a child to draw a large circle on

of paper. This is a question above and beyond the one pointed out (rightly) by Bartsch and Vennemann concerning how the reference set is inferred from the context, especially extra-sentential context.

We avoid some of the difficulties caused by the necessity of of having predetermined reference sets by making use of _functors_. According to Cresswell (1973), a _functor_ is a symbol which, occurring as the first member of a sequence of symbols of certain syntactic kinds, makes a sequence of the same or another syntactic kind. I am using this very broad and general interpretation of functor in this discussion. The _typical value_ functor applied to a concept with some measure attribute returns a value, e.g. the typical value of _size_ for _man_. Note that this is not the same as the typical man's size. The typical man's size is not readily determinable since it is hard to ascertain exactly what constitutes a typical man. A typical value functor is shown in Figure 4.7. We can abbreviate the typical value functor in a manner analogous to the collasping of predicates in the abbreviated network notation.

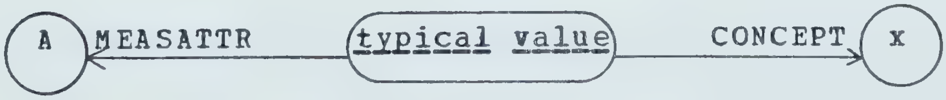


Fig. 4.7. A typical value functor.

Descriptive adjectives are treated as conjoined predications in most cases. The sentence "Judy ate a circular yellow cake" is diagrammed in Figure 4.8 (abbreviated

notation). Yellow in this sentence is treated as a kind of cake like a chocolate cake.

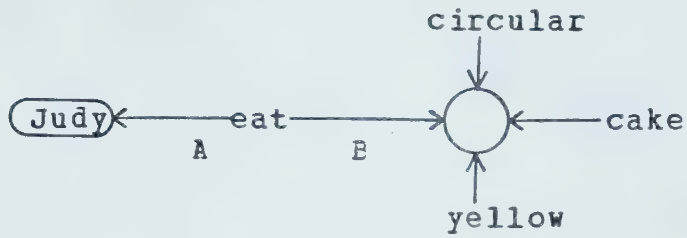


Fig. 4.8. "Judy ate a circular yellow cake".

Most adjectives appear to be comparative in nature regardless of their morphology. For example, big, tall, heavy, and so on are relative adjectives based on some measurable attribute of the object of focus. The sentence "John is a bigger than Bill" is diagrammed as shown in Figure 4.9.

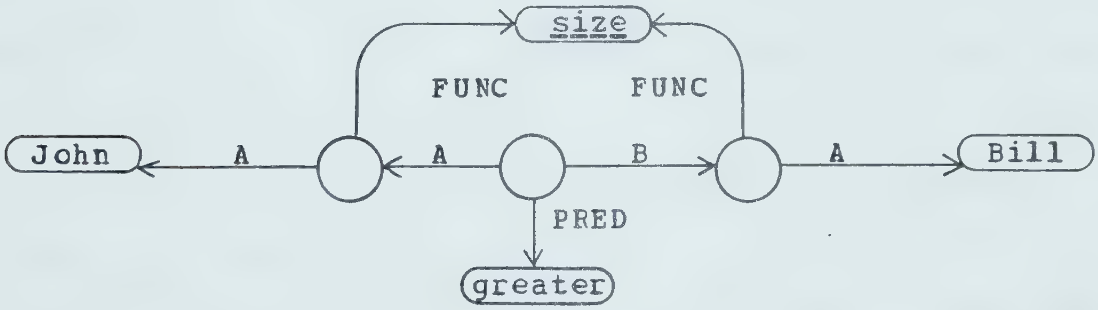


Fig. 4.9. "John is bigger than Bill".

The explanatory paraphrase of Figure 4.9 is "John's size is greater than Bill's size". Often the comparative is implicit in the utterance. For example, in the sentence "John is a big man" the adjective "big" serves as a comparative. The meaning of "John is a big man" is diagrammed as Figure 4.10. The associated paraphrase is as follows "John is a man and the

size of Jchn is greater than a typical value of size for a man". Notice that we are making use of the typical value functor in this example.

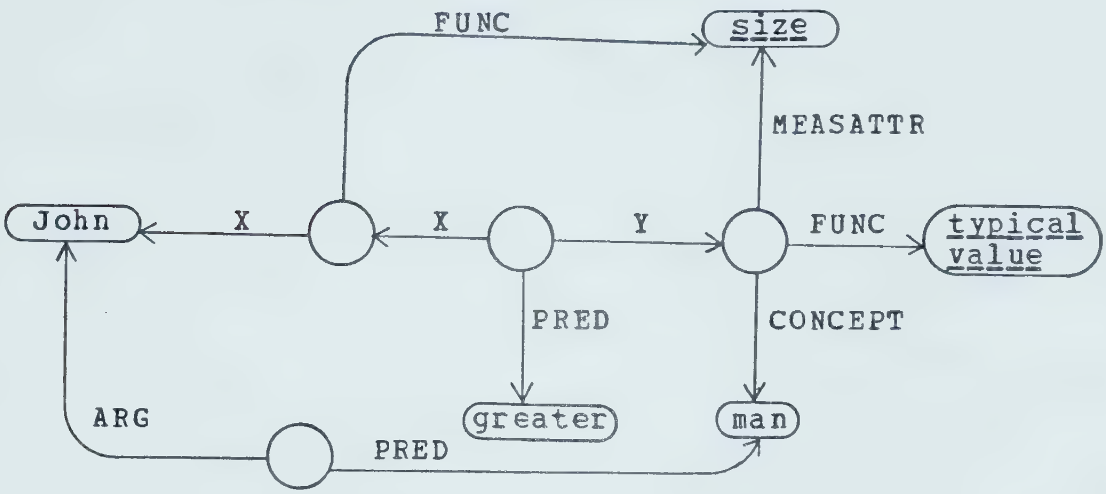


Fig. 4.10. "John is a big man."

One final use of a non-directly predicative, non-directly comparative use of a relative adjective of they type under discussion is shown in Figure 4.11. The sentence "Big John drinks the whiskey" is paraphrased in the diagram as "John drinks the whiskey and John's size is greater than the typical value of size for something and John is that something". In Figure 4.11, the node immediately to the left of John represents John's size (size is used as a functor in the proposition containing John and size). The treatment of relative adjectives based on measurable attributes, such as big, heavy, tall, etc. can be readily summarized as follows: The value of the "attribute" of "x" exceeds the value of the "attribute" which is typical for that concept (of which x is

an instance).

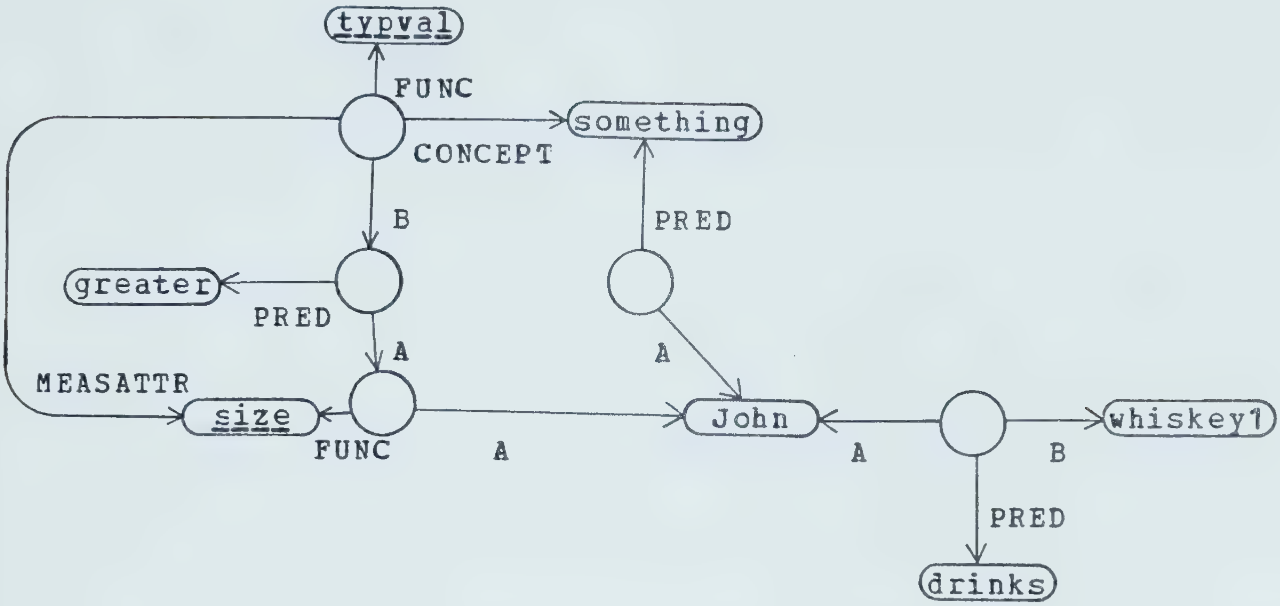


Fig. 4.11. "Big John drinks the whiskey".

Ordinary discourse admits constructions such as <4.7> through <4.10>.

John is the *perfect* man. <4.7>

Mary is the *worst conceivable* baker. <4.8>

Mike is the *ideal* fat man. <4.9>

In order to form a *more perfect* union... <4.10>

Modifiers such as *perfect*, *worst conceivable*, and *ideal* are problematic to represent because of the way they operate on what they modify. For example, we might formulate <4.7> in logical terms as:

$$(AP)\{[(Ax)[man(x)\&P(x)\square\!\!\Rightarrow y\text{-}approves[P(x)]]]\Rightarrow P(John)\} \quad \langle 4.7a\rangle$$

where y is the speaker. The formulation reads "John has all properties such that y would approve of any man's having

them". We can then easily formulate an expression for "someone is not a perfect man" by utilizing <4.7a> with the existential quantifier added (Ez)¬ and replacing P(John) with P(z). Clearly, the method of handling comparative adjectives such as big, tall, etc. does not work here.

At this time I make no definite proposals for handling adjectives such as perfect, ideal, worst kind of, best conceivable, etc., at any <u>detailed level of analysis</u>. However at a more superficial level level of analysis the sentence "John is a perfect host" is rendered as shown in Figure 4.12.

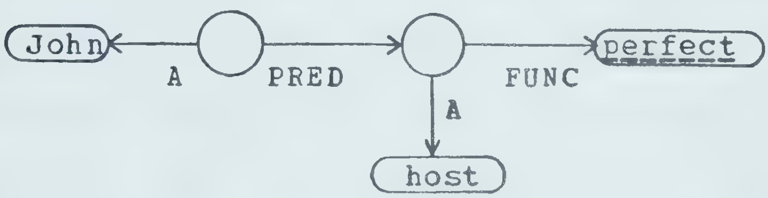


Fig. 4.12. "John is a perfect host."

If we later find out that "John is a basketball player", this information can be added to the structure shown in Figure 4.12, see Figure 4.13.

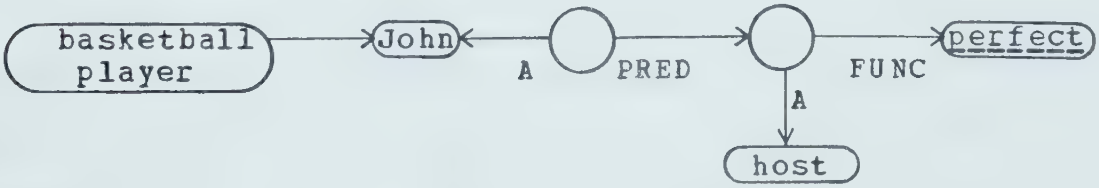


Fig. 4.13. "John is a basketball player and a perfect host".

4.5 Adverbial Constructions

The purpose of this section is twofold: first, to draw attention to the major problem of representing adverbial meanings; and second, to suggest some plausible methods for handling adverbial constructions within the state-based conceptual framework.

There appear to be two major approaches to the treatment of adverbial modifiers. One approach, expressed in Montague (1970), Bartsch and Vennemann (1972), Zadeh (1972), and others, is to regard comparative adjectives and adverbs as operators which transform predicates. The other approach, due to Reichenbach (1947), and seemingly accepted by Schank (1972) and Anderson and Bower (1973), is to regard adverbial modifiers as second-order predicates that impose constraints on a specific relation, thereby restricting the class of specific relations of which it may be a member. In particular we consider Bartsch and Vennemann's (1972) approach, which seems promising but will be seen to have serious defects.

Schank (1972) diagrams adverbs as action modifiers without further analysis. Apparently he has not concerned himself with the meanings of genuine manner adverbials so far (however, see Schank, 1974, for a discussion of adverbs such as vengefully, thoughtlessly, etc.). In the case of many adverbs (as in the case of many adjectives) this neglect is probably justified, since most of the meaning content derives

from perceptual processes. For example, in the sentence "Mary walked gracefully" it is difficult to paraphrase "gracefully" in more elementary terms. Essentially we know gracefulness when we see it. Thus perceptual understanding needs to be supplemented only by a few additional facts for language comprehension purposes, such as the fact that graceful motion is generally pleasing, is more or less the opposite of awkward motion, is smooth and well-coordinated, and the like. Other adverbial modifiers, however, clearly require systematic analysis; "quickly" is a good example. This term appears to say something about the speed of an action or activity, comparing it to some standard. An adequate meaning representation for "quickly" should spell this out precisely.

Bartsch and Vennemann (1972) suggest that adverbial modifiers operate on verb meanings in the same manner that adjectival modifiers operate on noun meanings, i.e. they have semantic representations with functors f such that f is applied to term x to map x onto a new term f(x). One problem in this approach is best illustrated by the following example.

John owns a large car. <4.11>

John is running quickly. <4.12>

Whereas large in <4.11> has as a reference set the set of cars, and John's car is large in relation to the "average" for that set, "running quickly" cannot be analyzed so easily. If the analogy were perfect then the reference set operated on by "quickly" would be the set of "runnings" (whatever that

means); but clearly this set of runnings must be further restricted to the set of runnings John is capable of performing. Thus "quickly" appears to operate not on "running" alone, but on "John running". As further examples consider <4.13> and <4.14>.

The cheetah is running quickly. <4.13>

The ant is running quickly. <4.14>

Clearly "quickly" here operates on "running ant" and "running cheetah" respectively.

Thus the nature of the "runner" is being used to narrow the reference set to which we apply a measure function. In <4.12> "quickly" modifies running with respect to John's runnings, or, if we don't know John, at least to human runnings (assuming that John is human). In <4.13> and <4.14> the measure function is applied to the runnings of cheetahs and ants respectively. Unfortunately factors other than the identity or category of the runner can also affect the meaning of "quickly", as shown by <4.15> to <4.19>.

John is running quickly on his hands and knees. <4.15>

John is running quickly on the moon. <4.16>

John is running quickly in Chile. <4.17>

The cheetah is running quickly in the dense forest. <4.18>

The cheetah is running quickly on the plain. <4.19>

The effect of locale on the meaning of "quickly" is seen in the contrast between <4.16> and <4.17> and between <4.18> and

<4.19>.

Thus it appears to us that the context which determines the meaning of an adverbial modifier cannot be circumscribed once and for all. In general, adverbials must be allowed to interact with any specific and general knowledge available about the participants in (and setting of) an action. In the approach of Zadeh (1972) to the treatment of adverbial "hedges" he specifies (weighted) components of each fuzzy term on which a hedge may operate <u>once and for all</u>. Because he needs to specify these (weighted) components prior to using a particular hedge, his approach lacks generality. In our semantic network, we would represent <4.13> without the adverb as diagrammed in Figure 4.14.

(a) (b)

(a) (Ex){(Ay)[cheetah(y) & ?(y) <=> x=y] & running(x)}

(b) (Ex){running(x)&cheetah(x)&?(x) & (Ay)[cheetah(y)&?(y)=>x=y]}

Fig. 4.14. "The cheetah is running."

(a) and (b) are based on alternative (but equivalent) representations of definite descriptions. Figure 4.15 then shows an attempt to represent the adverbial construction in <4.18> in keeping with Bartsch and Vennemann's (1972) general

approach but taking into account the above considerations. In the representation we show the explicit relationship between the speed of the cheetah's running as compared to the typical value of speed for something that is running, a cheetah, and in dense forests.

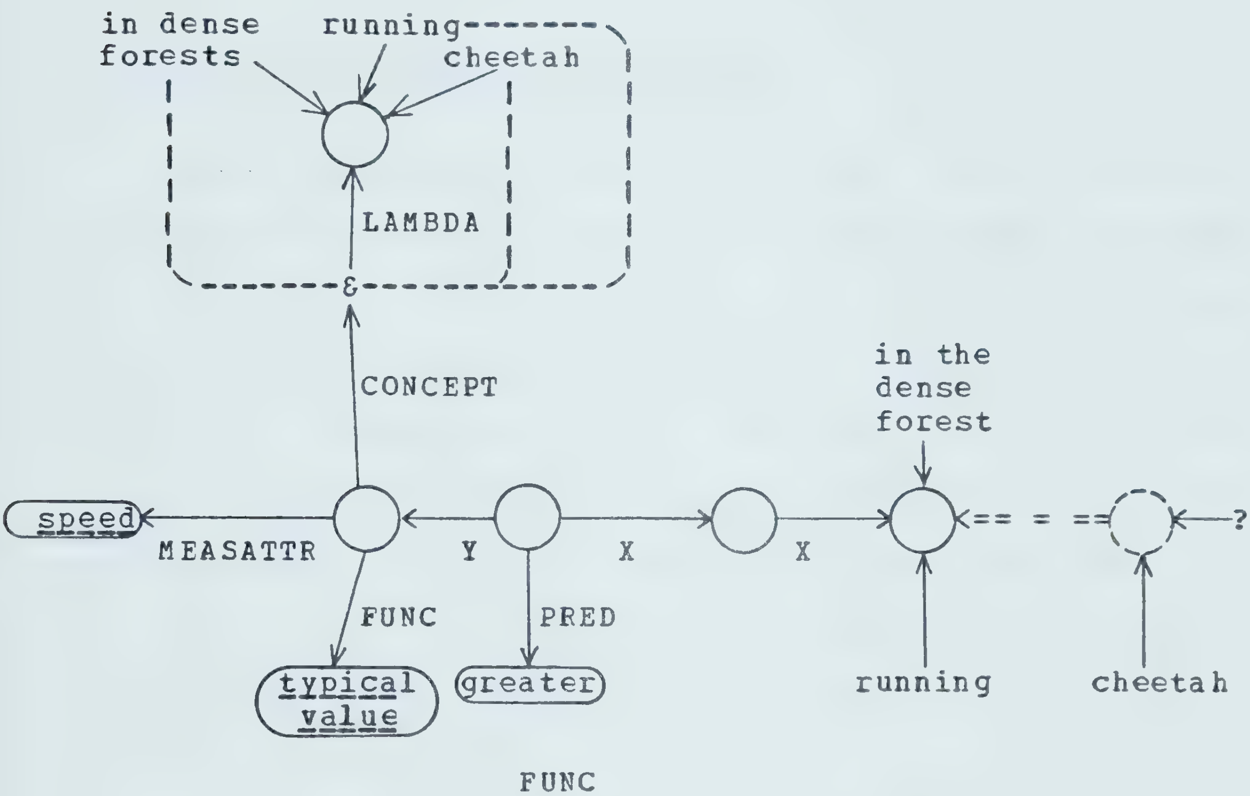


Fig. 4.15. "The cheetah is running quickly in the dense forest"

It is well to note that the set of cheetahs running in dense forests, required for comparison, may well be empty (if not, replace "dense forest" with "deep snow"). The "reference set" therefore, if it exists at all, is not of this world but of some imaginary world which is our conception of how hard cheetahs would find the going if they were to run through

forests (or snow). In our formulation we have applied the <u>typical value</u> functor to the lambda abstracted predicate

(LAMBDAx)[cheetah(x)&running(x)&in-dense-forests(x)].

The <u>typical value</u> functor does not presume the existence of a reference set.

4.6 Opacity and Vagaries of Reference

Certain linguistic forms give rise to referentially opaque contexts.[16] This is true of the <u>propositional attitudes</u> "believes that ...", "knows that ...", "wants to ...", and others, as well as other modalities created by causal situations, intentions, and the like, and conditional statements including the counterfactual conditional. <u>Quotation</u> creates referentially opaque contexts. While "simpleton" may be referentially equivalent to "fool", the statement <u>"simpleton" has nine letters</u> does not allow substitution of "fool" for "simpleton".

An example taken from Moore (1973) illustrates how a referentially opaque context can block existential quantification.

"John wants to marry a blonde." <4.20>

gives rise to two possible interpretations:

---

[16] An opaque context is one which does not allow substitution of referentially equivalent expressions or does not allow existential quantification. This is a well known problem discussed at length by Quine (1960,1961) and in Artificial Intelligence by McCarthy and Hayes (1969) and Moore (1973).

"John wants to marry a specific girl who also happens to be a blonde."; and <4.20a>

"John has no particular girl in mind, but he wants whoever he does marry to be a blonde." <4.20b>

The first interpretation, the transparent reading, can be existentially quantified, i.e. there exists someone whom John wants to marry. The second interpretation cannot be quantified in like manner since it contains an assertion about an existential statement rather than being an existential statement.

Various (equivalent) explanations have been given for the type of ambiguity shown in <4.20>. Philosophers tend to represent this as <u>scope</u> ambiguity of an existential quantifier. Some linguists however, prefer to represent the ambiguity as a distinction between a referential and attributive use of a noun phrase (see Partee, 1972). With the notation suggested by Schubert (1974) the opaque reading of <4.20> would be represented as shown in Figure 4.16.

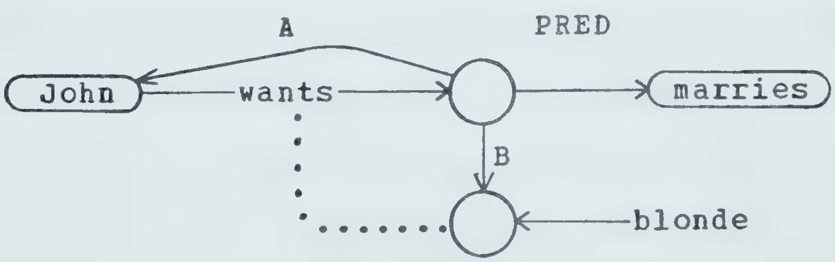


Fig. 4.16. "John has no particular girl in mind, but he wants whoever he does marry to be a blonde."

The transparent reading would be represented by Figure 4.16 if

we took out the dotted line.

Both Montague (1970) (as reported in Bartsch and Vennemann, 1972) and Lewis (1972) have developed theories that enable both the transparent and opaque readings for sentences like <4.20> to be generated. This is not carried far enough. The important problem remaining is how to choose the correct interpretation in <u>context</u>. This problem is investigated further in Chapter 6 and a crude scheme proposed to help choose the correct context based on prior knowledge.

In this Chapter we have taken a critical look at Schank's (1972) conceptual dependency theory and Wilks' (1973a) preference semantics theory. In doing so, we exposed many problems and suggested plausible methods for dealing with some of them. In particular, the basic representation in terms of states and events, the definition of complex concepts (most importantly action concepts), the handling of adjectives and relative terms as well as adverbial modifiers were examined. Solutions to the problems that these topics present to language processing systems have been developed to the point that some of them have been incorporated into the experimental program that supports this research.

Chapter 5

MEMORY AND LEXICAL STRUCTURE

5.1 Memory

The discussion of section 2.4 served to illustrate in a general way the role played by memory in processing language. Most of the elements that comprise memory were introduced in Chapter 3. Later in this Chapter some detailed examples of memory structures that represent the meanings of words concepts are shown. In this section the organisation of memory is considered. Some ideas are also presented that illustrate ways in which various heuristic techniques can be incorporated as an aid in memory processing.

The basic structural unit in memory is the proposition. The construction and interconnections of propositions comprise memory. In the past, predicate calculus has been used to represent propositional information adequately; and in Chapter 3 methods were outlined that showed how propositional information could be expressed adequately in semantic networks.

Schank (1975) has stated that "Since it is possible to say the same thing in any number of different ways, it is unreasonable to suppose that people are constantly checking to see whether a proposition that they have stored in one way in

memory is the same as another that they have stored somewhere else in a different form".[18] Schank then goes on at length discussing the need for a canonical form for meanings and proposes conceptual dependency as that representation. Nonetheless the type of representation used in conceptual dependency is a graphical form of predicate calculus. Using methods outlined in the previous two Chapters in no way restricts the development of many and varied heuristic techniques for organising the propositional information (which is what Schank has done); in fact this is desirable.

The formalism introduced in Chapters 3 and 4, along with the lexical structure described later in this Chapter form the basis for representing states, events, belief structures, counterfactuals, and the like. Nevertheless heuristics must be developed to provide an efficient computer implementation. The following ideas can (hopefully) be combined with the propositional base and prove to be a valuable aid in the formulation of a more extensive natural language understanding system.

In discussing memory organisation Schank (1975) further remarks "We can see at once that [hierarchical] organisation will not work for verbs or for nouns that are abstract or for

---

[18] Having noted that inference form the basis for understanding (inferences being propositions that have been left out of an utterance), it is not so unreasonable to suppose that the recognition of synonymy also requires inference.

nouns that do not submit easily to standard categories (such as teletypes)". Nevertheless "drinks" can surely be thought of as a subconcept of "ingests" and while some categories (teletype) would be difficult to fit into a _single_ hierarchy, they certainly can be fitted into various hierarchies (machine, transducer, communications system, etc.). We can easily impose a hierarchical (subconcept-superconcept relation) structure on top of the general concepts in memory (as shown in Figure 5.1) as a heuristic device.

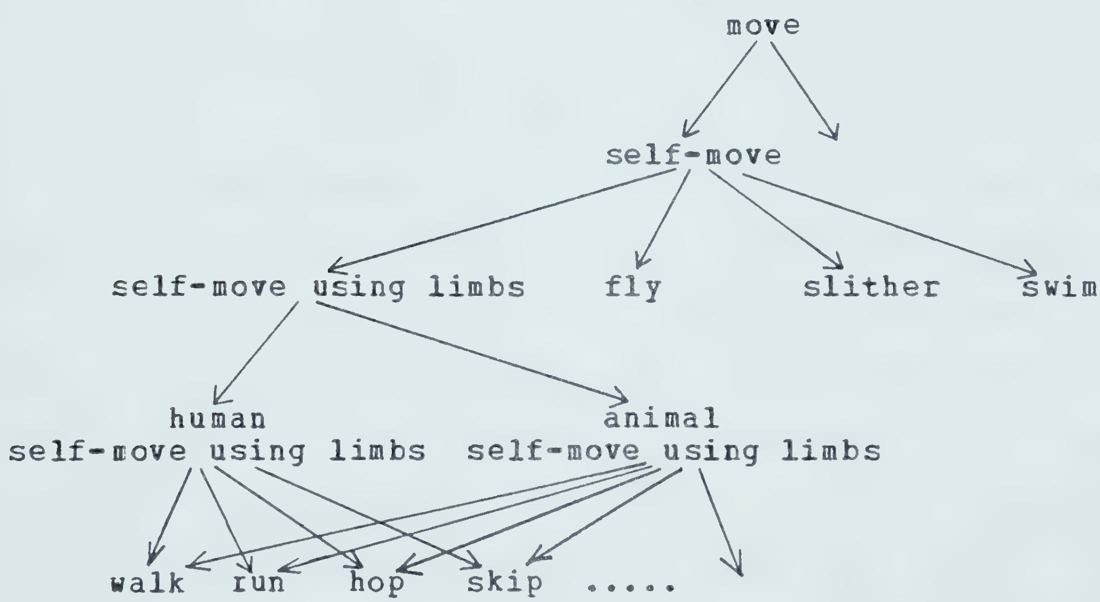


Fig. 5.1. Superimposed Hierarchical Structure

Currently _scripts_ (Schank, 1975; Abelson, 1973), _demons_ (Charniak, 1972), and _frames_ (Minsky, 1974; Winograd, 1974) are receiving a good deal of attention. They all have basically one thing in common; they can be thought of as a collection of facts and procedures associated with and clustered about a concept. This chunking of information around

a single concept seems to be at least a start in the right direction for the kind of understanding necessary for any successful computer use of natural language.

The following example from Winograd (1974) illustrates the use of frames. Associated with each frame is a set of important elements. Parts of the frames for "give" and "pay" are shown (simplified) below.

```
GIVE isa ACT
     ACTOR: person
     BENEFICIARY: person
     OBJECT: physical object

PAY isa GIVE
     OBJECT: money
     REASON: debt
```

GIVE is a kind of ACT, PAY is a kind of GIVE. The important elements are ACTOR, BENEFICIARY, OBJECT, and REASON. Among the many possible facts about a concept, only certain ones are relevant to a given situation, e.g. paying someone usually involves giving that person money in compensation of a debt you owe him. These relevant facts are the important elements. In addition, each frame has procedures associated with it. If we are trying to decide whether a particular act is a payment or not, a general procedure might cycle through the important elements seeing if they could be filled in appropriately for the situation. The procedure would be attached to the pay frame to help make plausible inferences.

Hierarchical structures have been used with success (Quillian, 1968) while frame systems are currently under

development at several AI laboratories. Further development of frame systems should prove helpful in answering some types of questions, e.g. the type Charniak (1972) uses as examples in his work with children's stories and piggy banks. Nevertheless questions like "Are elephants grey?" pose problems in organising knowledge. We might explicitly predicate "grey" of elephants but the thousands of items known about elephants would make the retrieval of relevant information to answer questions about elephant's color unrealistic. We might try to place "grey" under a subsuming concept of "appearance" which is one of several groupings of propositions about elephants. Perhaps in the near future frames or some variant of frames will provide an answer. For now, this remains an open question.

5.2 The Organisation of the Lexicon

By way of introduction, let me point out that the next two sections are distinguished for explanatory purposes only and there is no intent to dichotomize syntax and semantics. It is a rather touchy issue to decide whether, in fact, some of the features and category items used in the next section are syntactic at all.

5.2.1 Lexical Categories and Syntactic Features

Lexical items are items of vocabulary, usually, but not necessarily, words. Traditionally (the Aristotelian view), they are said to have both lexical (material) and grammatical (formal) meaning. This distinction can best be expressed in terms of open or closed classes (sets of alternatives) as explained below.

Lexical categories are properties associated with lexical items used in parsing. Through categories, the representation of an appropriate lexical item can be selected from the lexicon. Normally, categories are identified with classes rather than expressing a particular sytagmatic relation among items in an utterance. Normally, classes are separated into open classes and closed classes. Typically, closed classes have a strictly limited membership which cannot be increased by adding new formations or loanwords (words which have been incorporated by one language from another language). The significance of closed class items is best expressed by their grammatical function. In contrast, open classes have a large, readily increasing membership. New formations and loanwords are easily integrated. The meaning of open class words is best expressed through synonyms. The dichotomy between open and closed class words represents a mixture of criteria, both statistical (the number of forms in a class), and diachronic (concern with the way in which

language changes over time).

The lexical categories for English shown in Table 5.2.1.1 are an adaptation of the categories used by Woods et al. (1972) and Winograd (1972). The lexical items have been classified according to class and feature (features shown in Tables 5.2.1.2 and 5.2.1.3). Some of the decisions made in this classification were arbitrary, especially those pertaining to whether a word or group of words should form a new class or be given a new feature within a class. However, since the classification scheme is used to aid in parsing and not as a constraint, we need not be unduly concerned with this type of arbitrariness. For example, whenever a word classified as a binder (BIND) is encountered in text, it serves to seperate clauses. Other categories help in selecting possible actions, AM's, comparatives, NM's, etc.

OPEN CATEGORIES

N	nominal, typically either a noun (man, airplane, city) or a proper noun (John, IBM)
A	action, typically a verb (walk, throw, fly)
NM	nominal modifier, typically an adjective (tall, happy)
AM	action modifier, typically an adverb (quickly, suddenly)

CLOSED CATEGORIES

CONJ	=	conjunction (and, or, but)
BIND	=	binder (before, if)
PREP	=	preposition (to, for, over)
PRO	=	pronoun (I, you, they)
DET	=	determiner (the, a, those)
ORD	=	ordinal (first, second, last, final)
NEG	=	negative (not)
COMP	=	comparative (more, less, greater)
OP	=	operation (plus, times)
QWORD	=	question nominal (who, what, why)
QNTFR	=	quantifier (some, any, none)
PRT	=	particle (knock out)
NUM	=	number (one, two, three)
INTJ	=	interjection (oh)

TABLE 5.2.1.1 - Lexical Categories

The syntactic features that can be attached to the various lexical items are shown in Tables 5.2.1.2 (open category) and 5.2.1.3 (closed category) and explained in greater detail below. The syntactic features are necessary to insure formal agreement in person, number, gender, or tense between two or more lexical items or parts of sentences.

OPEN CATEGORIES

NOMINAL's (N)

NS	singular	TIME	time
NP	plural	FTIME	functional
COLL	collective	PERS	personal
POSS	possessive	DIM	diminutive

ACTION's (A)

AUX	auxiliary	BE	be
WILL	will	DO	do
HAVE	have	MODAL	modality
TRANS	transitive	ITRNS	intransitive
PART	participle	IREG	irregular
PRES	present	PAST	past
INF	infinitive	TPS	3rd person

NOMINAL MODIFIER's (NM)

ADJ	adjective
COM	comparative
SUP	superlative
CLASF	classifier

ACTION MODIFIER's (AM)

AT1	adverb type one
AT11	adverb type one one
AT2	adverb type two
AT22	adverb type two two
AT3	adverb type three
AT4	adverb type four
AT5	adverb type five
AT6	adverb type six
AAA	adverb modifying another adverb or adj
AA	adverb modifying an action
AP	adverb modifying a prep or prep phrase
ADT	adverb specifying definite time
AIT	adverb specifying indefinite time
AL	adverb specifying location
AJ	adverbial adjunct

TABLE 5.2.1.2 - Syntactic Features (open categories)

The syntactic features that can be attached to nominals include number, either NS (e.g. flower), NP (e.g. flowers), or COLL (e.g. acreage). Number agreement is necessary between

nouns and associated determiners. Some nominals possess all three number features such as "fish". PERS is another nominal feature that indicates a personal nominal (e.g. employee). DIM is attached if the nominal is a diminutive (e.g. booklet). POSS indicates possession as in "John's". The time features differ according to whether the nominal is a time word (e.g. day, year - TIME) or indicates a time (e.g. yesterday - FTIME).

Actions have perhaps the most complex set of features. Typically verbs, they have irregular forms and need features to indicate tense, voice, use, transitivity, number, and mood. The feature AUX indicates the use of the verb as an auxiliary (ie. a verb form used in forming the tenses, moods, voices, etc. of other verbs). Included in the auxiliaries are the features BE, DO, HAVE, WILL, and MODAL.[21] These features help in determining constituents of action phrases. Other features include TRANS or ITRNS which determine whether something is affected by the action (e.g. they saw a light - TRANS) or not (e.g. do not lie - ITRNS); INF denotes the infinitive; PAST and PRES denote tense; and PART denotes the participle. IREG is a feature that points out an irregular verb form. A large

---

[21] They include auxiliaries of periphrasis, which assist in expressing the interrogative, negative, and emphatic forms of speech, viz. do (did); auxiliaries of tense, have, be ,shall, will; of mood, may, should, would; of voice, be; of predication (i.e. verbs of incomplete predication which require a verbal complement), can, must, ought, need, also shall, will, may, when not auxiliaries of tense or mood. (OED s.v. Auxiliary, B. Sb., 3.)

amount of English verb information can be garnered from morphology. Normally verbs have one of the following forms: the root, which can be used as the infinitive, present tense, subjunctive, or imperative; the root plus "ing", which can be used as a gerund or present participle; the stem plus sibilant suffix, which is used as third person singular of the present tense; and the root plus dental suffix, which is used as the past participle or simple past tense.

Nominal modifiers have only four features, i.e. ADJ, COM, SUP, and CLASF. Respectively, they stand for adjective (e.g. big), comparative (e.g. bigger), superlative (e.g. biggest), and classifier. Classifiers may be other nominals used as classifiers as in Winograd's (1972) example, "water meter cover adjustment screw".

The type features attached to AM's are specific to adverbial modifiers. In part, adverbial modifiers are based on Zadeh's (1972) work, especially the adverbial modifiers with features AT1 and AT2. Features classify adverbs according to how they operate in an utterance. The feature AT1 is attached to adverbs that act as transformation operators (Zadeh, 1972) on single fuzzy sets as as in "John was very decent".[22] The type one adverb "very" would uniformly raise the degree to which qualities contributing to decency are present in the

[22] The type features have all been placed under the category AM due mainly to the nature of action modifying adverbs. However as type one adverbs show, this is not always the case.

subject of the predication. The feature AT2 applies when the adverb acts as an operator on a predicate in the following way. In "John was essentially decent", "essentially" operates on decent to accentuate those aspects of decency which are most crucial to its possession and de-emphasizes those features which are less crucial. The features AT11 and AT22 are similar to AT1 and AT2. They indicate more context dependence; the effect of AM's with these features is partially determined by their proximity to the verb they modify, for example the adverb "slightly" as well as some sentence adverbials (classified AT11). The feature AT3 applies to predicate limiting adverbs such as "emotionally" and "healthwise". Manner adverbs, e.g. quickly, quietly, etc., have the feature AT4 attached. Whenever an adjective acts as an adverb of degree it is given the feature AT5, as in "I was dead tired". Finally AT6 is the applicable feature for modal adverbs, such as "certainly", "possibly", and so on.

Other features that can be attached to AM's include AAA, AA, AP, ADT, AIT, AL, and AJ. AAA applies to adverbs like "very" that can modify other adverbs or adjectives. An AA modifies verbs; the adverbs "quickly", "quietly", and "easily" are examples. AP's are adverbs modifying prepositions or prepositional phrases as in "directly under the window". Adverbs specifying definite time include "now" as in "Where are you living now" and have attached the feature ADT. Adverbs specifying indefinite time have the feature AIT attached and

include "soon" and "also", for example "we soon found out his address". AL's specify locations as in "put it there". AJ's are adjuncts such as "nowhere", "never", "nevermore", etc., for example "he was nowhere to be found".

CONJ	BIND	PREP	NUM
ORD	OP	PRT	NEG
COMP	QWORD	INTJ	

PRO's

NP	plural	COLL	collective
NS	singular	POSS	possessive
REL	relative	PERS	personal
DEM	demonstrative	DEF	definite
INDEF ...	indefinite	SUB	subject
OBJ	object		

DET's

DEF	definite	INDEF ...	indefinite
NP	plural	COLL	collective
NS	singular	DEM	demonstrative
QDET	question		

QNTFR's

NE	negative	NS	singular
NONUM ...	no number	NP	plural
COLL	collective		

TABLE 5.2.1.3 - Syntactic Features (closed categories)

The syntactic features attached to closed category lexical items are as follows. For the categories CONJ, BIND, PREP, NUM, INTJ, ORD, OP, PRT, NEG, QWORD, and COMP there are no additional features.

The features that apply to determiners (DET) are the following: INDEF for indefinite determiners such as "a" and

"an" or QDET's such as "which" and "what"; DEF for definite determiners such as "the" or DEM for demonstrative determiners such as "this", "that", "those", and "these"; and the number features NS, NP, and COLL.

Pronouns have features similar to nominals, i.e. number features NS, NP, and COLL. They can be either definite, DEF, usually replacing a previously mentioned or inferred nominal, e.g. his, or indefinite, INDEF, e.g. anyone. They can be possessive, POSS, e.g. mine, his; personal, PERS, e.g. I, us; demonstrative, DEM, e.g. they, he; interrogative or relative, REL, e.g. who, whose; and they can be used as subject, SUB, e.g. he, or object, OBJ, e.g. him.

Quantifiers can be NE (negative) like "no" or "none" or they can be those quantifiers that cannot be combined with numbers such as "many" - NONUM. Quantifiers also possess the number feature for number agreement with quantified entities - NS, NP, and COLL.

5.2.2 Meaning Representations for Word Concepts

Associated with open class category words are meaning representations: one for each sense of the word. The structure of a meaning representation is based on the semantic network notation developed in Chapter 3. Pragmatic and semantic information are included in a meaning representation for words.

Figures 5.2 through 5.7 show networks that illustrate some of the main senses of the word drink, concentrating on action aspects. For illustrative purposes Figures 5.2, 5.4 and 5.7 are divided into a pragmatic section and a semantic section. The pragmatic section includes the template(s) that guides the parse of the utterance and two lists: the first list contains propositions that represent the implications that are likely to be needed for the comprehension of subsequent text; and the second list contains propositions representing critical implications that we expect to match in the surface structure. In Figure 5.2 this first list is (P3) and the second list is (P1,P2). The semantic section contains the network that represents the meaning of the word sense. Figures 5.3, 5.5, and 5.6 show various nominal senses of the word "drink".

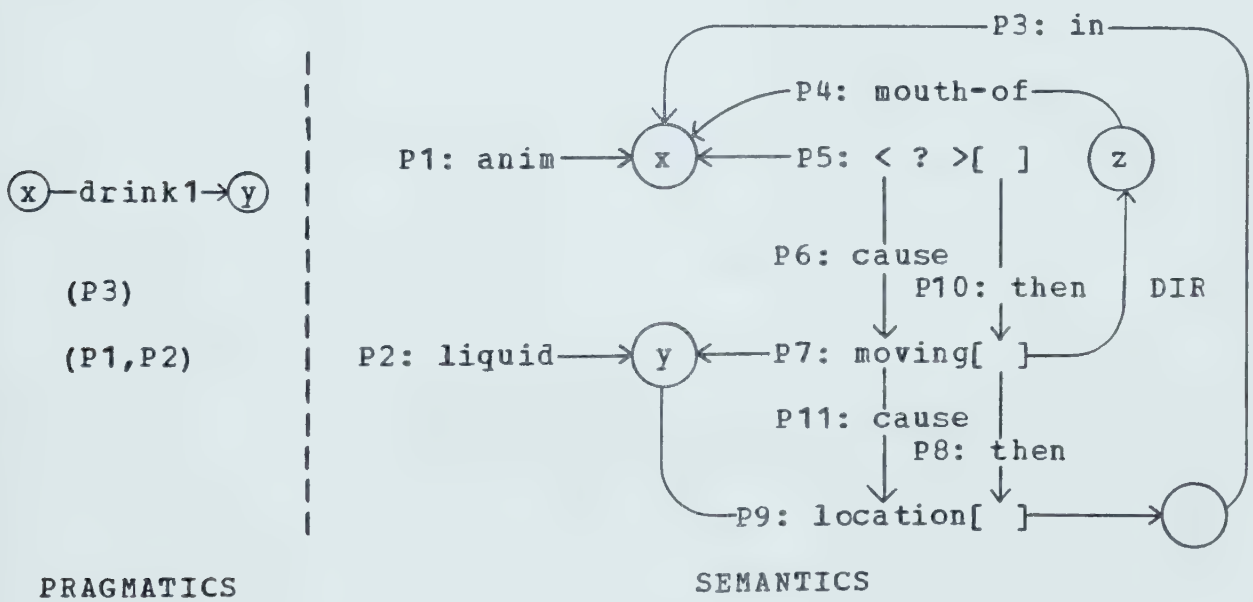


Fig. 5.2 "(John) drinks (water)"
"(Mary) drinks (prune juice)"

Fig. 5.3 "(John is a) drinker"

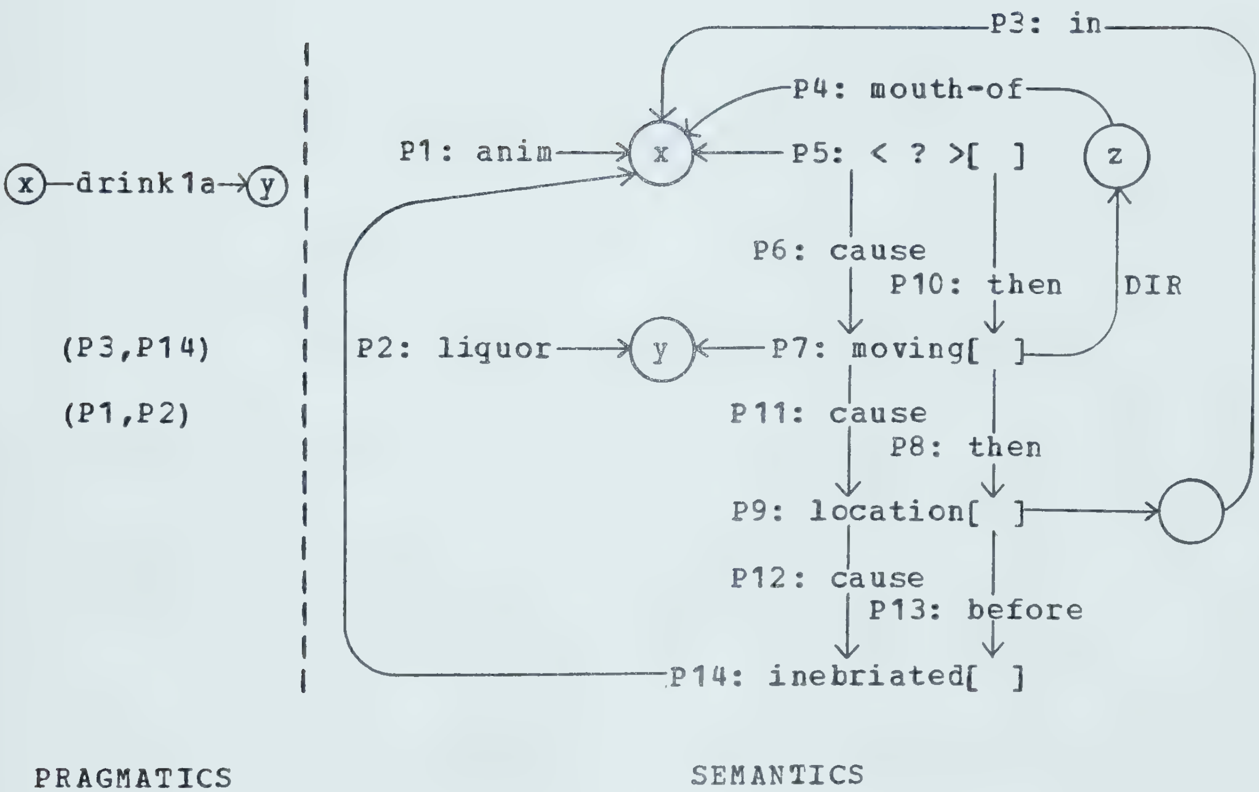


Fig. 5.4 "(John) drinks (whiskey)"
"(John) drinks"
"(Mary has a) drinking (problem)"
"(Mary) drinks (a lot)"

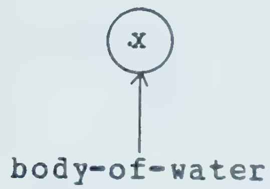


Fig. 5.5 "(Throw John into the) drink"

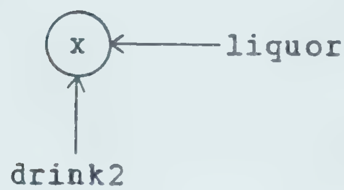


Fig. 5.6 "(John is drinking a) drink"

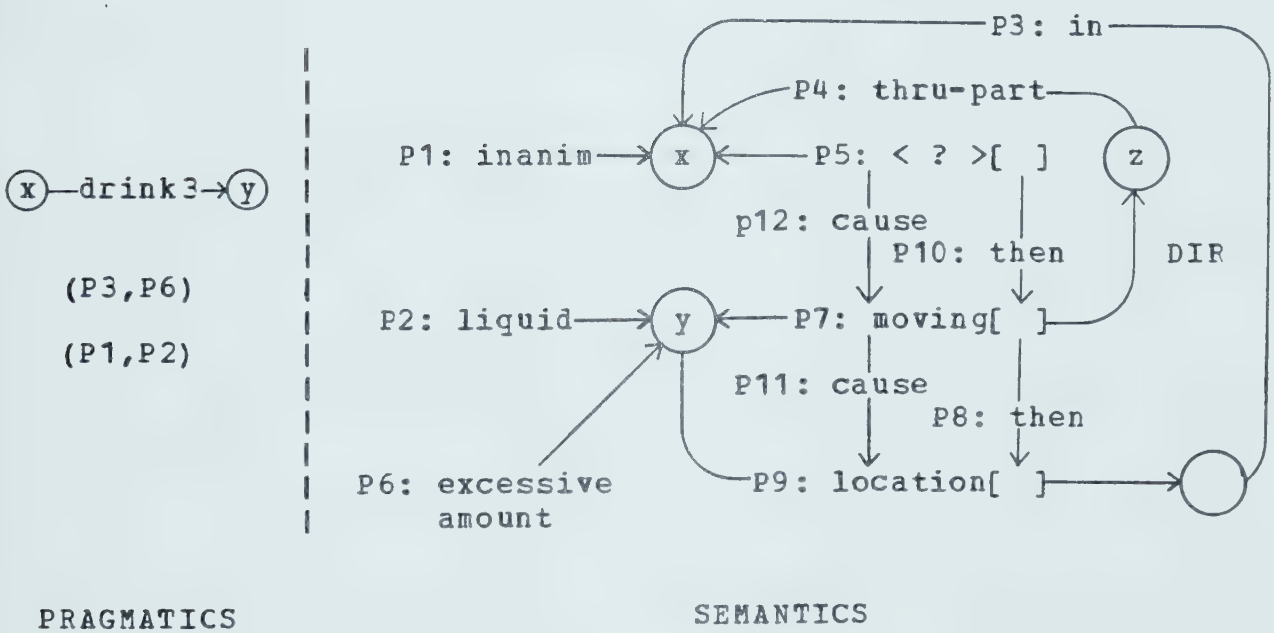


Fig. 5.7 "(My car) drinks (gasoline)"
"(The donut) drinks (coffee)"
"(The rag) drinks (spills)"

Notice that Figures 5.2, 5.4, and 5.7 all have the notion of <u>change in containment location</u> in common. This corresponds to a <u>general concept</u> that subsumes not only differing senses of "drink" but also other more specific concepts as well, like "eating" or "receiving an enema". This observation has led to the following consideration.

When creating the meaning representations (networks) for concepts it is desirable to avoid the duplication of propositions in storage. If we extract more general concepts from the specific concepts that they subsume (totally or in part), we can avoid duplication by associating the common propositions with the more general concept.

In a sense the work of both Schank (1972) and Wilks (1973a) supports the contention that the meaning of a concept is best represented by predications at the highest level of generality that adequately explain the term's meaning. Thus we extract from "drinking" (and eating, etc.) the structure shown in Figure 5.8.

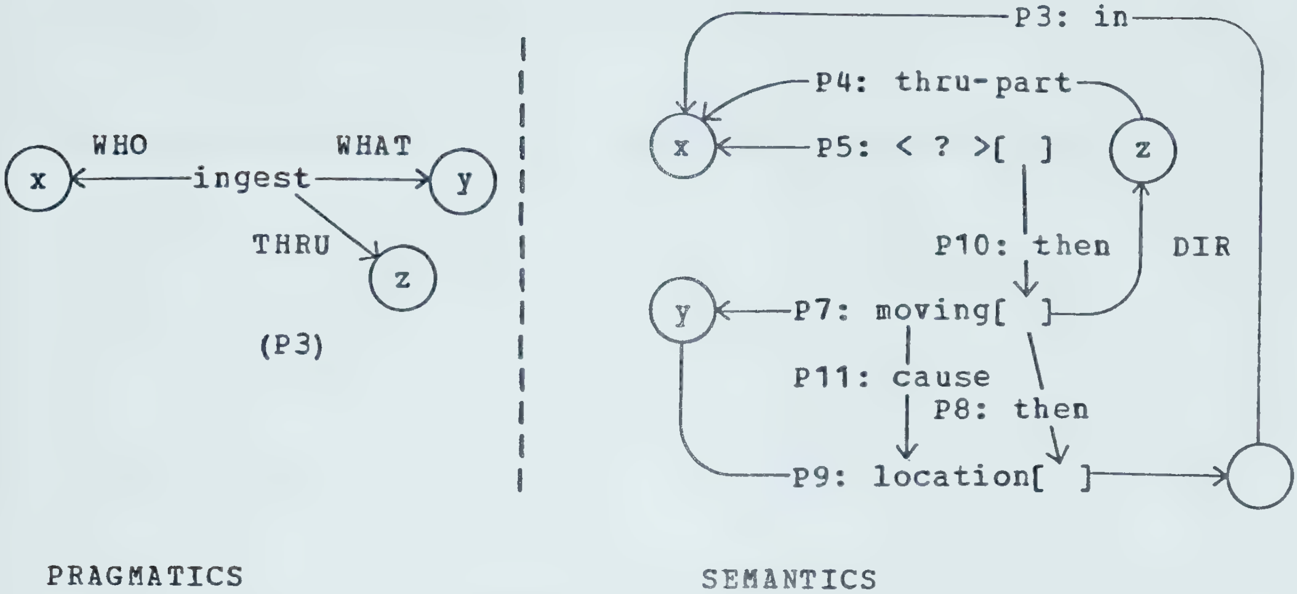


Fig. 5.8 "ingest"

We might reasonably label the concept expressed by this structure "ingest". It is important to note, however, that while Schank and Wilks might conclude that "ingesting" is a primitive action, I consider it a general concept. This applies to all primitive actions put forward of Schank and Wilks. Examination of Figure 5.8 shows clearly that ingesting is <u>not a primitive</u> action but one whose meaning is expressed in terms of causes, motion, time, and other concepts.

At this point the original representations for the various action senses of "drink", i.e. Figures 5.2, 5.4, and 5.7, can be replaced with more simplified diagrams based on the general concept "ingest". Figure 5.9 shows the representation of "drink" as expressed in Figure 5.2 redrawn in terms of the general concept "ingest". In similar fashion Figure 5.10 diagrams one meaning of "eating", again based on

the general concept "ingest".

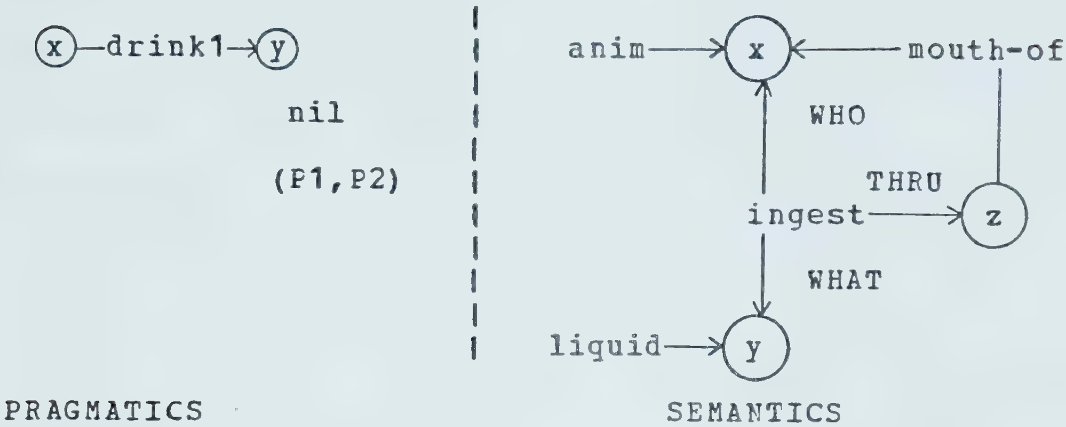


Fig. 5.9 "(John) drinks (water)"

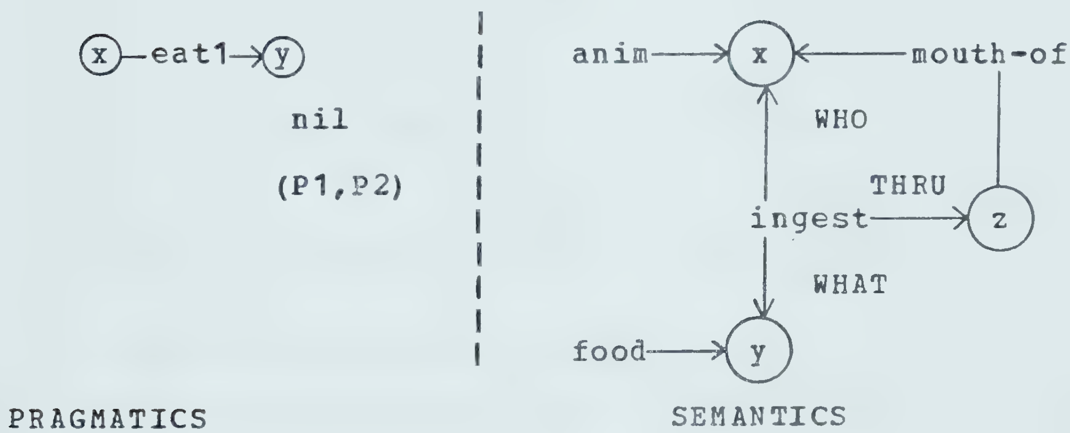


Fig. 5.10 "(John) eats (cake)"

The key to making effective use of the meaning representation for comprehension centers on the propositions that contain arguments that we expect to match in the surface utterance. The lexical item for "drink" would contain, among other things, pointers to a list of propositions; these propositions contain the arguments that we expect to match with words in the text and are most frequently needed for

comprehension. At times, however, other propositions may be required for comprehension. For example, the word sense illustrated in Figure 5.2 shows that we expect to find, in an utterance about drinking, an anim(x) and a liquid(y), propositions P1 and P2. But the question can be posed, "What is the effect of John's drinking". To answer this question would entail a further investigation of the other propositions in the network, especially the first list of implications. Although it is implicit in the semantic structure, we make explicit in the pragmatic structure the inference that "x - drink - y" necessarily implies that it causes y's location to be <u>in</u> x at some time after x initiates the drinking action. Of course, since this implication is common to all senses of "drink" (and eats, inhales, etc.) it is abstracted into the same general concept "ingest" as well, as shown in Figure 5.8.

The semantic structure for each word sense for "drinks" is represented as properties attached to the word sense. The main properties include ARGS, the argument list containing arguments used in the word sense; IMPLICS, a list of implications that accompany the word sense; the propositions P1, P2, etc. that relate the arguments and predicates that make up the network explicating the given word sense; and templates of the form

arg1 arg2 ... argi WORD argi+1 ... argn

The implications make the most commonly used inferences part of the meaning representation of a word concept. The

propositions, for example P1 and P4 shown in Figure 5.2 are, in turn, represented as shown in Figure 5.11. See Appendix B for sample lexical entries, in particular the entry for "drink".

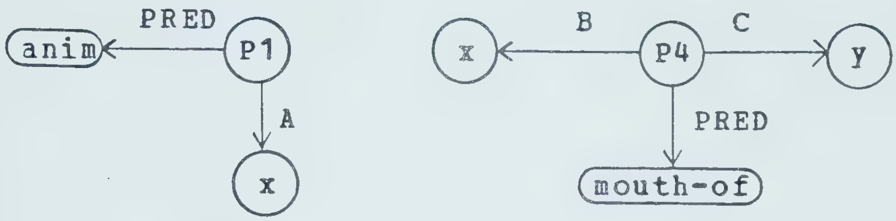


Fig. 5.11

Many advantages accrue by representing meaning formulas in this way. First, unlike Wilks' (1973a) meaning formulas, the representation is suggestive of the meaning of a word. I see no justification for (binary) lexical decomposition trees as meaning representations for words as such trees are neither suggestive of the type of processing required nor of the propositions they encode.

A second and major advantage is this. The meaning representation for a word is not required to be explicitly in terms of "primitives". Rather, each of the predicates in the propositions that form the network representing the meaning of the word can, in turn, be represented in an analogous manner. In particular the notion of a "cause" seems to me to be no more "primitive" than "drink". This method of representing word meanings enhances the representational schema for the purpose of comprehension since any amount of detail can be

included in the meaning representations by adding propositions to the networks.

Third, inference mechanisms, heuristic processing algorithms, and superimposed knowledge-organizing schemas can be incorporated using this representation for word meanings as easily as in any other representation. Incomplete information in surface text can be inferred, when necessary, directly from the meaning representation, in some cases as a missing argument.

The use of this type of meaning representation for lexical items is further explained in Chapter 6.

5.2.3 Formal Specification of Lexical Items

Table 5.2.2 shows the syntax according to which lexical items are put into the dictionary. The notation used is basically the Backus Normal Form (BNF) metalanguage with the addition of the Kleene * operator. The metalinguistic characters include brackets < >, Kleene operator *, the form ::=, and the bar |. Brackets surround phrase class names which stand for sets of entities. The form ::= can be read as 'is of the form'. The bar is used to denote alternation, one form or the other. And the * is used to define an arbitrarily repeatable (zero or more) constituent when surrounded by brackets otherwise it is used to define an arbitrarily repeatable (one or more) constituent; e.g. <a*> means zero or

more a's while a* or <a>* means one or more a's.

```
<lexical entry>        ::=  ( <root> <meaning *> )
<root>                 ::=  root of word given meaning
<meaning>              ::=  <lexical category>
                              <category value> <synonym>
                              <antonym>  <compound>
                              <idiom>  <abbrev>
<lexical category>     ::=  <open category> |
                              <closed category>
<open category>        ::=  N  |  A  |  NM  |  AM
<closed category>      ::=  CONJ  |  PREP  |  ...
<category value>       ::=  < <root feature list>
                               <word sense formula *> >*
<synonym>              ::=  SYN <synonym value> * | <null>
<antonym>              ::=  ANT <antonym value> * | <null>
<idiom>                ::=  ID <idiom value> * | <null>
<abbrev>               ::=  ABB <abbrev value> * | <null>
<compound>             ::=  COMPD <compound value> *
                            | <null>
<synonym value>        ::=  a synonym for the root
<antonym value>        ::=  an antonym for the root
<idiom value>          ::=  the idiom or slang expression
                              involving the root
<abbrev value>         ::=  abbreviation for root
<compound value>       ::=  (   <tree> *  )
<tree>                 ::=  ( <word> <result> <tree> * )
<result>               ::=  <word>  |  <null>
<word>                 ::=  any word
<null>                 ::=
<root feature list>    ::=  (<morph code><root feature*>)*
<word sense formula>   ::=  ( the construction of semantic
                             units and/or concepts that
                             express the correct word
                             sense | a function to be
                             applied )
<morph code>           ::=  -ING | -ED | ... |  <null>
<root feature>         ::=  AT1 | DEF | BIND | ...
```

TABLE 5.2.2 - Syntax for Lexical Items

Chapter 6

PROCESSING NATURAL LANGUAGE

Much research into natural language understanding has neglected theory in favour of performance. Problem domains were restricted and theoretical considerations were sacrificed for programming considerations. Extensibility has been implausible or, at best, highly dubious. A cohesive theory of natural language understanding is necessary; one that makes it easy to determine the semantic structure that represents easily understood utterances. The theory is then embodied in the structural rules with computer programs assuming the role of supporting and validating the theory.

6.1 Introduction

Research reported by Schank (1972, 1973) and Wilks (1973a, 1973b, 1973c) has greatly influenced this thesis. Though I believe their approaches to be lacking in several important respects, some of their techniques are impressive. The most important of these are perhaps their emphasis on events, de-emphasis of syntax, and interpretive directness, as discussed in Chapter 1.

I have claimed elsewhere (Cercone and Schubert, 1974) that there is a level of representation that is "semantically deeper" than either Wilks' or Schank's (1973) systems admit

and that this level of representation is necessary for comprehension. Mcre generally, the representation needs to be flexible to accomodate understanding at various levels, i.e. whatever level is appropriate for a given situation. The representation described above has this flexibility, and, at the same time, allows application of various heuristic techniques such as those used by Schank and Wilks.

The semantic structure expressing a particular utterance is formed according to simple structural rules. The central role of verbs is acknowledged in the experimental program and preferred semantic categories for the subjects and objects of verbs are used to guide each choice in the creation of meaning structures. Word sense disambiguation for not only verbs but modifiers and nominals also, follows naturally in this approach and is explained in detail in section 6.2. Extensive trial and error searches are eliminated since the interpretation of utterances takes on a "slot and filler" character. The approach to interpretation is almost completely semantically oriented and syntax is used only when meaning analysis fails. The rest of this Chapter describes an experimental program designed to build semantic structures, as described in the previous three Chapters, from natural language utterances.

6.2 Interpretation of Natural Language Text

Building structures, like those shown in the previous three chapters, from unprocessed natural language text is the main topic of this Chapter. Except as noted, computer programs have been written to do all that is discussed in this section. Sample outputs from these programs appear in the Appendices.

6.2.1 Initial Classification

Initially text is read (either in discourse mode or from an external file for longer text) and broken into clauses (at present this process is very unsophisticated). Each clause is then classified in the following manner. Words in clauses are morphologically analysed and, based on that analysis, they are classified to determine all of their possible syntactic functions in an utterance. For example, the form "drinks" of the root word "drink" can only be used nominally or as an action (see also Appendices A and C). The root form is located in the lexicon (see Appendix B for a more complete discussion) and using affix information from the morphological analysis, all of the possibilities for the word in question are extracted from the lexicon. The sample listing preceding Figure 6.1 gives the results of this classification scheme, clause by clause, under the heading +++ THE CLASSIFIED UTTERANCE IS +++. When all of the words in a clause are classified in this manner, the next phase, parsing, begins.

6.2.2 Parsing

Traditionally, the object of parsing sentences has been to output syntactic trees. These trees served as input to semantic routines charged with the generation of meaning structures. Winograd (1972) and Woods (1970) tried, with some degree of success, to integrate the two processes and use each of them to guide the other. Schank (1972) and Wilks (1973) have stressed that syntactic processing was secondary to meaning analysis and should be necessary only when the resolution of ambiguity by meaning analysis alone had failed. The parsing phase is almost completely semantically oriented. One important by-product in the method to be described is the detection of the correct sense of nominals and actions and, although not yet implemented, modifiers as well.[21]

The parsing proceeds as follows. Words, in a clause that has been classified, are scanned from left to right in search of a suitable candidate for an action. Once found, the sentence is seperated into

((FIRST PART) (ACTION CANDIDATE) (SECOND PART))

The action candidate contains, among other things, a list of possible action senses that this particular root form may have. These senses are ordered by a scheme, albeit a very superficial scheme, to be described later. Associated with word senses are templates; they have been described in Chapter

---

[21] I am restricting utterances to active voice.

5. For example the sense *GIVE1 of the root form "give" has a template

X GIVE Y Z

and an alternative (ALTERN) template

X GIVE Z TO Y

associated with it.

The template, e.g. "X GIVE Y Z", is used to guide the parsing. In this example X, Y and Z are variables representing the arguments of the predicate "give" that we expect to find in the surface utterance in the given order.[22] More detailed information concerning the arguments is obtained by examining the network propositions, for the sense of "give" in question, that involve the arguments. Thus X, in this case, would represent an ANIMATE nominal capable of "giving".

This is very similar to what Schank does when parsing in conceptual dependency theory. If the words in the surface utterance do not satisfy the constraints for arguments of the predicate being examined, it is due to one of four reasons. First, alternate syntactic constructions could exist. Second, a different <u>sense</u> of the action is "correct". Third, the particular action candidate in question is not the action of the clause. Finally, some other reason, like slang expressions might be the cause.

---

[22] In the event an argument is not present in the utterance, the implication template can be used for inference purposes to infer arguments.

Whenever arguments fail to satisfy predicates, a search for alternative implication templates begins. The result of this search is shown quite clearly in Figure D10 of Appendix D for the ternary predicate "give". In that example "give" is used syntactically in two different forms to distinguish the indirect object, one with the preposition TO and one without. If this approach fails then the list of senses for the root form is further examined. If other senses of the action candidate exist, they are examined further to see if arguments of the action candidate in the surface utterance match variables in the template. This procedure is repeated until the correct sense of the action candidate is found or the list of senses is exhausted. If the sense list is exhausted, scanning continues in the surface clause for another suitable action candidate and the process is repeated.

Part of the process of matching arguments of predicates in surface text to variables in implication templates involves finding the correct sense of nominals and modifiers as well. The sentence "A drinker drinks many drinks" (shown in Appendix D as Figure D6) has as the second argument of the predicate "drinks" the word "drinks". Possible nominal senses for that "drinks" include an alcholic beverage, a body of water (throw John into the drink), or a thirst quencher. Thus, if the first sense of a nominal fails as argument, all other senses must be examined before deciding not to accept it as argument. This reasoning applies with respect to modifiers in a similar but

not identical fashion. For instance, a "yellow cake" is a type of cake much like a chocolate cake whereas a "yellow car" is something that is yellow and something that is a car. Using these methods, sentences such as "A *drinker* *drinks* many *drinks* " and "The pilot *banked* his plane near the river *bank* over the *bank* that he *banks* on for good *banking* service" present little difficulty.

Morphological analysis is important since only those forms that can authentically be considered as actions need be examined. In the example, "A drinker drinks many drinks" the word "drinker" is eliminated immediately as an action candidate due to morphological analysis. Thus, we are very quickly able to get a right choice for an action candidate.

Both Schank and Wilks used their intuition to set up their respective meaning representations. The way that they defined and used semantic "primitives" are one example. One way in which my intuition has shaped the experimental program can be shown with the following superficial scheme for choosing word senses. Consider the following structure:

```
                          "bank"
  ┌─────────────────────┬─────────┴──────────┬─────────────────────┐
(g1,11)              (g2,12)              (g3,13)              (g4,14)
(16,0)               (92,0)               (47,0)               (12,0)
```

Associated with each sense of a word are gi's and li's which denote frequency counts for *global* and *local* usage of the ith meaning sense of the word. Whenever the term "bank" is encountered in text, the local frequency counts are first

examined to see if any context has been established in the dialog thus far. Since they are all zero, assuming the initial values shown above, no context has as yet been established and the global frequency counts are examined. Accordingly the second sense is selected as the most likely candidate for the meaning of "bank". If it fails then the third, first, and fourth senses would be selected in that order. Suppose that the third sense turns out to be the correct sense for the term "bank" in this case. The local frequency count is set to one and whenever the term "bank" is encountered the third sense will be selected first and its local frequency count will be incremented by one (if it is the correct sense). This would continue until the third sense fails to be the correct sense. At this point we would examine the second, first, and fourth senses in that order until we arrive at the correct meaning sense of "bank" (i.e. the ith term). l3 is then added to g3, li is set to one (nonzero), l3 is reset to zero, and the ith meaning sense is selected whenever the term "bank" is encountered and so on.

The sample listing preceding Figure 6.1 gives the results of the parsing phase, clause by clause, under the heading +++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++. The list of modifiers found in the clause, further classified as to function, and associated with correct predicate arguments they modify, is also given as part of the parsing phase. For the most part these modifiers (if any) have been

associated with their corresponding argument based on proximity to the argument.

6.2.3 Building the Semantic Network

Once the parsing phase has been completed the meaning representation is built for the clause and that structure is integrated into the semantic network. The first step involves building an intermediate structure based only on the predicate of the clause and its arguments. After this intermediate structure is created, it may be altered to accomodate other information detected in the parsing phase. This information includes mainly modifiers (only certain of the adjectival modifiers are analysed, however adverbial and quantificational are planned).

Presently five classes of adjectival modifiers are analysed. These include predicatives (John is short), adjectives (Bill has a yellow car), explicit comparatives (Bill is heavier than John), implicit comparatives (Big Mary ate a large steak), and some functors (Joe is a perfect cook). In addition, modifiers can be combined in various ways and be correctly analysed. Some examples are given in Appendix D, most notably, "Short, small Anne is eating the delicious, light, round, yellow cake" (Figure D.5) and "Big Mike is a perfect fat man" (Figure D.7).

The order in which modifiers are handled can greatly

affect processing time. If we were to analyse "Big Mike" as a perfect man and then as a perfect fat man we would not only take more time but probably construct an incorrect meaning representation. It would be correct to construct the predicate fat man and then operate on that construction with the functor perfect. In the experimental system the order in which modifiers are analysed and integrated into the semantic network is roughly the following: predicatives, adjectives, implicit comparatives, functors, and explicit comparatives.

The following is an example of the experimental program's performance. Three simple sentences (without modifiers) are classified, parsed, and integrated into a semantic network using the techniques just discussed. Note well that descriptive phrases like "that man" and "the coffee" are treated attributively (Russell's analysis of descriptions). Nevertheless we still wish to single out a previously introduced referent, without assuming that a referring description defines the referent fully. The method is to add an "unspecified property" to the properties supplied in the description of the referent, to indicate that the description contains some but not all items of information needed to distinguish the referent from all other entities in the universe. Note that the incomplete definite description would be used not to insert a new node in semantic memory but to access (through pattern-matching against recent referents, say) an already existing node. Thus "unspecified properties"

do not appear in semantic memory, but only in the immediate semantic transform of an utterance.

```
R NEW:MACLISP
15:22.35
(RESTORE 'CHKPT)
    NIL
(UNDERSTAND)
    READY

JOHN IS DRINKING THE WHISKEY. THAT MAN
DRINKS THE COFFEE. THAT MAN DRINKS THE WHISKEY.


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((N (NS ((0 0) (/*JOHN1)))) (A (IREG PRES ((0 0) (/*EQ1)
    ))) (N (NS ((0 0) (/*DRINK6))) A (PART ((0 0) (/*DRINK1
    P1 P2)) ((0 0) (/*DRINK1A P1 P2)) ((0 0) (/*DRINK3
    P1 P2))) NM (ADJ CLASF ((0 0) (/*DRINK2 P1 P2)) ((0
     0) (/*DRINK4 P1 P2)) ((0 0) (/*DRINK5 P1 P2)) ((0
    0) (/*DRINK6 P1 P2)))) (DET ((NS NP COLL)) (/*THE))
     (N (NS ((0 0) (/*WHISKEY1)))))

    +++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

     ((*DRINK1A *WHISKEY1 Y) (*DRINK1A *JOHN1 X))

    +++MODIFIERS+++

    PREDICATIVES->
     NIL

         +++ THE SEMANTIC NET +++
      *ATOM*      *VALUE*     *PROPERTY*

    *WHISKEY1     PROP0002     PROPOSITIONS
    PROP0002     *WHISKEY1     PRED
    PROP0002     INST0003     ARG
    INST0003     PROP0001
                  PROP0002     PROPOSITIONS
    PROP0004     INST0003     ARG
    INST0003     PROP0004
                  PROP0001
                  PROP0002     PROPOSITIONS
    PROP0004     *UNS0005     PRED
    *UNS0005     PROP0004     PROPOSITIONS
    PROP0001     INST0003     Y
    PROP0001     *JOHN1     X
    *JOHN1     PROP0001     PROPOSITIONS
    PROP0001     *DRINK1A     PRED
```

```
*DRINK1A      PROP0001      PROPOSITIONS

+++ THE CLASSIFIED UTTERANCE IS +++

((PRO ((DEF DEM REL)) (/*THAT)) (N (NS ((0 0) (/*MAN1)
))) (N (NP ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))) A
(PRES TPS ((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A
P1 P2)) ((0 0) (/*DRINK3 P1 P2)))) (DET ((NS NP COLL)
) (/*THE)) (N (NS ((0 0) (/*COFFEE1)))))

+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*DRINK1 *COFFEE1 Y) (*DRINK1 *MAN1 X))

+++MODIFIERS+++


PREDICATIVES->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

*COFFEE1      PROP0007      PROPOSITIONS
PROP0007      *COFFEE1      PRED
PROP0007      INST0008      ARG
INST0008      PROP0006
               PROP0007      PROPOSITIONS
PROP0009      INST0008      ARG
INST0008      PROP0009
               PROP0006
               PROP0007      PROPOSITIONS
PROP0009      *UNS0010      PRED
*UNS0010      PROP0009      PROPOSITIONS
PROP0006      INST0008      Y
*MAN1      PROP0011      PROPOSITIONS
PROP0011      *MAN1      PRED
PROP0011      INST0012      ARG
INST0012      PROP0006
               PROP0011      PROPOSITIONS
PROP0013      INST0012      ARG
INST0012      PROP0013
               PROP0006
               PROP0011      PROPOSITIONS
PROP0013      *UNS0014      PRED
*UNS0014      PROP0013      PROPOSITIONS
PROP0006      INST0012      X
PROP0006      *DRINK1      PRED
*DRINK1      PROP0006      PROPOSITIONS

+++ THE CLASSIFIED UTTERANCE IS +++

((PRO ((DEF DEM REL)) (/*THAT)) (N (NS ((0 0) (/*MAN1)
```

```
))) (N (NP ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))) A
(PRES TPS ((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A
P1 P2)) ((0 0) (/*DRINK3 P1 P2)))) (DET ((NS NP COLL)
) (/*THE)) (N (NS ((0 0) (/*WHISKEY1)))))

+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*DRINK1A *WHISKEY1 Y) (*DRINK1A *MAN1 X))

+++MODIFIERS+++

PREDICATIVES->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

*WHISKEY1      PROP0016
               PROP0002      PROPOSITIONS
PROP0016      *WHISKEY1      PRED
PROP0016      INST0017      ARG
INST0017      PROP0015
               PROP0016      PROPOSITIONS
PROP0018      INST0017      ARG
INST0017      PROP0018
               PROP0015
               PROP0016      PROPOSITIONS
PROP0018      *UNS0019      PRED
*UNS0019      PROP0018      PROPOSITIONS
PROP0015      INST0017      Y
*MAN1      PROP0020
               PROP0011      PROPOSITIONS
PROP0020      *MAN1      PRED
PROP0020      INST0021      ARG
INST0021      PROP0015
               PROP0020      PROPOSITIONS
PROP0022      INST0021      ARG
INST0021      PROP0022
               PROP0015
               PROP0020      PROPOSITIONS
PROP0022      *UNS0023      PRED
*UNS0023      PROP0022      PROPOSITIONS
PROP0015      INST0021      X
PROP0015      *DRINK1A      PRED
*DRINK1A      PROP0015
               PROP0001      PROPOSITIONS

NIL
(MTS)
15:23.40
```

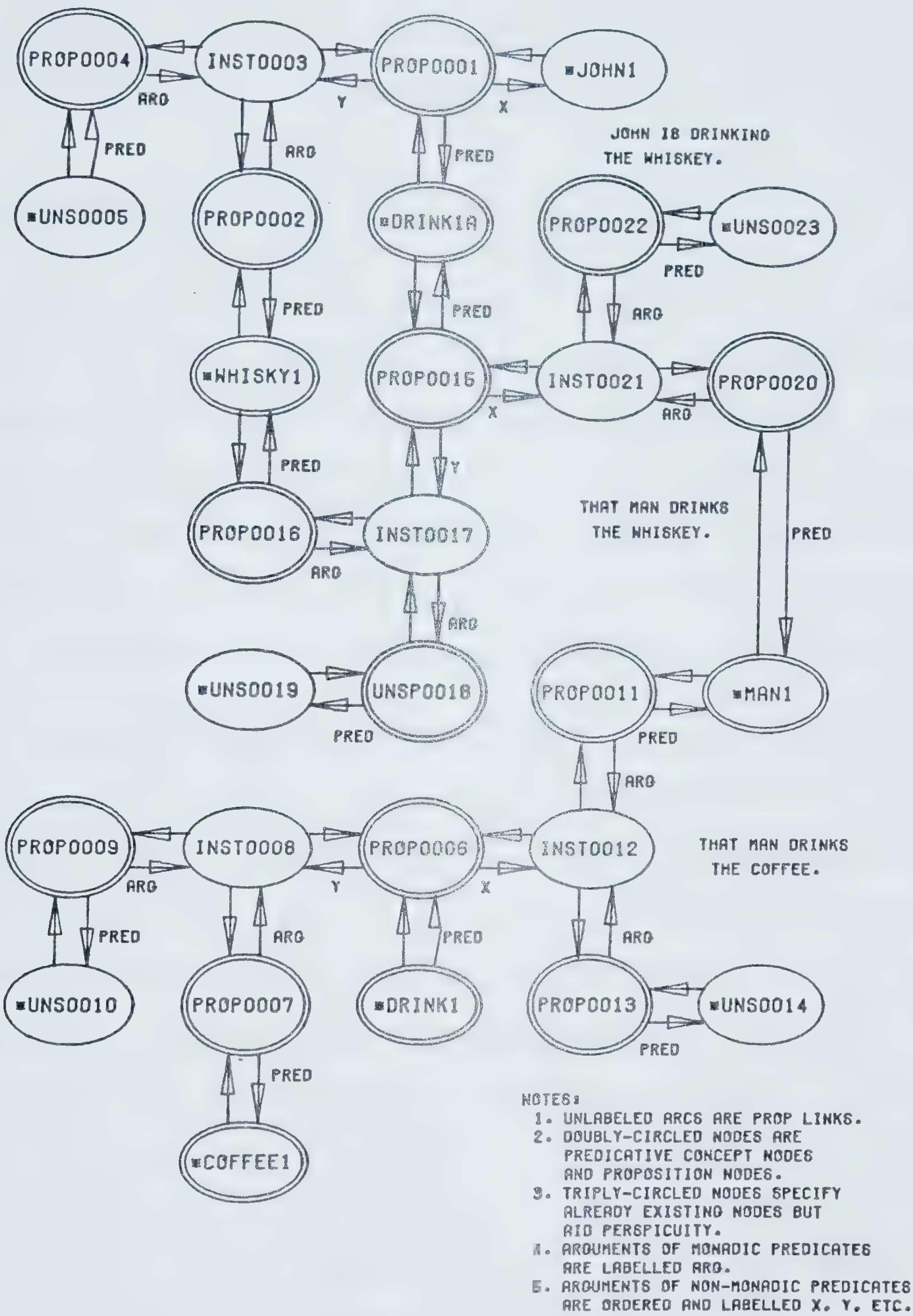


Fig. 6.1. A Very Simple Example

Chapter 7

CONCLUSIONS

7.1 Significance

The single most significant result of the research reported in this thesis is the sound theoretical basis established for further research into natural language understanding. This is not to say that all thought and speech can be represented adequately, but it is my hope that new problems, or as yet unsolved problems, will find themselves easily accomodated within the framework outlined above.

Specific proposals were made for representing logical and natural language quantifiers. These proposals are aimed at enhancing the expressive power of semantic networks; they may not be computationally optimal. Intentionality, which has proved problematic in many representations, is easily expressible; so are other propositional attitudes. The arbitrary (and unnecessary) distinction between actions and states, and between intentional action and purposive causation was examined in detail. Logical inadequacies that have appeared in other representations, especially inadequacies with quantifiers and connectives, were remedied.

On the practical side, many of the structures outlined have been created by an experimental program. A lexical

structure was devised to work quickly and efficiently for this program. The lexical structure also accomodates the meanings of complex concepts (as discussed in Chapter 4) in a systematic way. In allowing for complex concepts, the machinery was provided for a deeper level of comprehension.

However, this theory is not without its problems. Although lacking a full implementation of the present theory, major theoretical problems can be posed and these are discussed in the next section.

7.2 Problems

In setting up meaning representations for ordinary English discourse, we, as humans, have an unparalleled ability to solve three problems which have proved especially difficult to systematically automate. These are the problems of inference, reference, and modification. In this thesis some progress has been shown with modifiers (chiefly certain of the adjectival modifiers) and little progress with the problems of reference and inference.

There is no simple syntactic solution to the problem of finding the correct referents of referring expressions, such as "them" in Wilks' (1973a) example.

> The soldiers fired at the women
> and I saw several of <u>them</u> fall.

He remarks "In the case of someone who utters the 'soldiers and women' example sentence, what he is to be taken as meaning

is that the women fell. It is of no importance in that decision if it later turns out that the soldiers fell. What was meant by that sentence is clear, and not merely a likelihood matter".

The problems in dealing with inference are no less challenging. Whenever people communicate, a vast amount of information is assumed by the speaker and inferred by the listener. Communication would be all but impossible without this protocol. Schank (1975) gives an example of how important inference is to comprehension. His four examples are:

(1) John went to a restaurant. He asked the waitress to tell the chef to cook him a hot dog.

(2) John went to a restaurant. He asked the bus driver to talk to the midget.

(3) John went to a birthday party. First Bill opened the presents and then they ate the cake.

(4) John went to a birthday party. He asked the waitress to tell the chef to cook him a hot dog.

The first and third examples make sense to most users of English since all the objects prefaced by "the" are known implicitly through a user's knowledge of restaurants and birthday parties. It is usually incorrect in English to preface an object by "the" unless it is unique or has been previously introduced. The second and fourth examples violate common sense knowledge about restaurants and birthday parties.

In order for a listener to infer missing propositions he needs an adequate supply of contextual and world knowledge. Since inferences are never certain, part of this large problem

must be to determine methods that establish inferences that are most likely true and part of the problem must deal with organising knowledge in a way that allows inferences to be made efficiently.

The treatment of modifiers has a good start in this research but it remains to be seen whether the methods used for certain of the adjectival modifiers can extend to all classes of adjectival modifiers as well as other types (e.g. adverbial modifiers).

Understanding natural language is an extremely complicated endeavour. A large part of this endeavour must be aimed at solving the three major problems just mentioned.

7.3 Plans for Future Research

The experimental program, with relatively few heuristics, creates network structures on the average of about three seconds of CPU time per sentence. This is accomplished with an _interpreter_ LISP system running under the Michigan Time-Sharing system [MTS] on an IBM 360/67 computer. One of the first steps to be taken in the future will be to attempt a fuller implementation with a larger vocabulary within the question-answering framework. Certain features are desirable, namely data-base consistency, error analysis, feedback, and more complete inference mechanisms. However, not only the lighter side of building a system, the programming, but also

the theoretical side needs further development.

Future plans also include working on the problem of inference discussed earlier. Schank's (1975) scripts, Minsky's (1974) frames, and Charniak's (1973) demons, all approach this problem. I feel, in addition to these approaches, that scripts, frames, or demons could and are constructed on the spur of the moment by humans. Consider "a booze and skinny-dip party to celebrate the opening of a nudist colony". We can understand a story on such a theme even though we probably do not possess a ready-made "script" for it.

The problem of using descriptions to refer has been ignored in this implementation. Thus, descriptions such as "the man" are treated attributively in the Russellian sense. Yet it is incorrect to regard referential descriptions as disguised assertions. Schubert (1974) explains by example:

> Suppose that the language understanding system is told "John's car is red". The system would first look for an existing node to use as referent of "John's car". We need not be concerned with the details of this search here, noting only that if it succeeds, no new description is placed in memory. Only "red" is predicated about the node found (provided this predication is consistent with prior knowledge). If the search fails, however, the system creates a new existentially quantified node with the attached proposition that this is the one and only car John has... This Russellian existence assertion is placed in memory, provided it is consistent with prior knowledge.

In the more distant future, I would like to continue working on the modification problem and then the myriad of problems presented by inference.

REFERENCES

Abelson, R. P. (1973). "The Structure of Belief Systems", in <u>Computer Models of Thought and Language</u>, R. Schank and K. Colby (eds), W. H. Freeman, San Francisco, California, pp 465-471.

Amarel, S. (1968). "On Representations of Problems of Reasoning about Actions", in <u>Machine Intelligence</u>, D. Michie (ed), American Elsevier Publishing Company, New York, v 3, pp 131-172.

Anderson, J. and Bower, G. (1973). <u>Human Associative Memory</u>, V. H. Winston and Sons, Washington, D.C.

Badre, N. (1972). "Computer Learning from English Text", Electronics Research Laboratory, Memo No ERL-M372, University of California, Berkeley, California.

Bartsch, R., and Vennemann, T. (1972). <u>Semantic Structures</u>, Athenaum Verlag, Frankfurt, Germany.

Beckmann, P. (1971). "Computerization of English", <u>IEEE Spectrum</u>, v 8, n 12, pp 20-27.

Benjamin, C. (1939). "Science and Vagueness", <u>Philosophy of Science</u>, v 6, pp 422-431.

Black, F. (1968). "A Deductive Question-Answering System", in <u>Semantic Information Processing</u>, M. Minsky (ed), MIT Press, Cambridge, Massachusetts, pp 354-402.

Bobrow, D. (1968). "Natural Language Input for a Computer Problem-Solving System", in <u>Semantic Information Processing</u>, M. Minsky (ed), MIT Press, Cambridge, Massachusetts, pp 146-226.

Borko, H. (1967). <u>Automated Language Processing</u>, Wiley and Sons, New York, New York.

Bratley, P. and Dakin, D. (1968). "A Limited Dictionary for Syntactic Analysis", in <u>Machine Intelligence</u>, E. Dale and D. Michie (eds), American Elsevier Publishing Company, New York, New York, v 2, pp 173-181.

Bruce, B. (1972). "A Model for Temporal References and its Application in a Question-Answering Program", <u>Artificial Intelligence Journal</u>, v 3, n 1, pp 1-26.

Cercone, N. (1973). Unpublished Survey in Artifical Understanding and Question-Answering Systems. Course requirement in CS 665 - Seminar in Artificial Intelligence.

Cercone, N. (1974). "Computer Analysis of English Word Formation", Technical Report TR74-6, Department of Computing Science, University of Alberta, Edmonton, Alberta.

Cercone, N. and Schubert, L. K. (1974). "Toward a State-Based Conceptual Representation", Department of Computing Science, TR74-19, University of Alberta, Edmonton, Alberta.

Chafe, W. (1970). Meaning and Structure in Language, University of Chicago Press, Chicago, Illinois.

Charniak, E. (1969). "Computer Solution of Calculus Word Problems", Proceedings of the First International Joint Conference on Artificial Intelligence, Washington, D. C., pp 303-316.

Charniak, E. (1972). "Toward a Model of Children's Story Comprehension", AI TR-266, PhD Thesis, MIT, Cambridge, Massachusetts

Chomsky, N. (1957). Syntactic Structures, Mouton, The Hague.

Chomsky, N. (1965). Aspects of the Theory of Syntax, MIT Press, Cambridge, Massachusetts.

Chomsky, N. (1966). Current Issues in Linguistic Theory, Mouton, The Hague.

Clark, H. (1973). "Space, Time, Semantics, and the Child", in Cognitive Development and the Acquisition of Language, T. Moore (ed), Academic Press, New York, New York.

Cohen, P. (1974). "A Prototype Natural Language Understanding System", Technical Report No. 64, Department of Computer Science, University of Toronto, Toronto, Ontario.

Coles, L. (1972). "Techniques for Information Retrieval Using an Inferential Question-Answering System with Natural Language Input", SRI Project 8696, Technical Note 74, Stanford, California.

Collins, A., and Quillian, M. (1972). "How to make a Language User", in Organization of Memory, Tulving, E., and Donaldson, W. (eds), Academic Press, New York, New York.

Cooper, W. (1964). "Fact Retrieval and Deductive Question-Answering Information Retrieval Systems", Journal of Association for Computing Machinery, v 11, n 2, pp pp 117-137.

Copilowish, I. (1939). "Border-Line Cases, Vagueness, and Ambiguity", Philosophy of Science, v 6, pp 181-195.

Cresswell, M. (1973). Logics and Languages, Methuen and Co. Ltd., London, England.

DesCartes, R. (Ouvres philosophiques. Edited by L. Aime-Martin, Paris 1842). [The Philosophical Works of DesCartes , vols I and II. Translation by Elizabeth S. Haldane and G. Ross, Cambridge, 1911].

Davidson, D. and Harman, G. (eds) (1972). Semantics of Natural Language, D. Reidel Publishing Company, Boston, Massachusetts

Dewar, H., Bratley, p., and Thorne, J. (1969). "A Program for the Syntactic Analysis of English Sentences", Communications of the Association for Computing Machinery, v 12, n 8, pp 476-479.

Evans, C. (1967). "States, Activities, and Performances", Australasian Journal of Philosophy, v 45, pp 293-308.

Feigenbaum, E. And Feldman, J.(eds) (1963). Computers and Thought, McGraw Hill Book Company, New York, New York.

Fillmore, C. J. (1968). "The Case for Case", in Universals in Linguistic Theory, E. Bach and R. Harms (eds), Holt, Rinehart, and Winston, New York, New York, pp 1-88.

Findler, N. and Chen, D. (1971). "On the Problems of Time, Retrieval of Temporal Relations, Causality, and Co-existence", Advance Papers of Second International Joint Conference on Artificial Intelligence, The British Computer Society, London, England, pp 531-545.

Firschein, O. and Fischler, M. A. (1971). "A Study in Descriptive Representation of Pictorial Data", Advance Papers of Second International Joint Conference on Artificial Intelligence, The British Computer Society, London, England, pp 258-269.

Fodor, J. (1972). "Troubles About Actions", in Semantics of Natural Language, D. Davidson and G. Harman (eds), D. Reidel Publishing Company, Boston, Massachusetts

Garvey, T. and Tenenbaum, J. (1974). "On the Automatic Generation of Programs for Locating Objects in Office Scenes", Proceedings of the Second International Joint Conference on Pattern Recognition, Copenhagen, Denmark, pp 162-168.

Goguen, J. (1974). "Concept Representation in Natural and Artificial Languages: Axioms, Extensions, and Applications for Fuzzy Sets", International Journal of Man-Machine Studies, v 6, n 5, pp 513-552.

Goodman, N. (1951). The Structure of Appearance, Bobbs-Merrill Company Inc., New York, New York.

Green, B, Wolf, A., Chomsky, C., and Laugherty, K. (1963). "BASEBALL: An Automatic Question-Answerer", in Computers and Thought, E. Feigenbaum and J. Feldman (eds), McGraw Hill Book Company, New York, New York, pp 207-216.

Green, C. (1969). "Theorem Proving by Resolution as a Basis for Question-Answering Systems", in Machine Intelligence, B. Meltzer and D. Michie (eds), American Elsevier Publishing Company, New York, New York, v 4, pp 183-208.

Green, C. and Raphael, B. (1968). "The Use of Theorem Proving Techniques in Question-Answering Systems", Proceedings 23rd ACM National Conference, pp 169-182.

Guzman, A. (1971). "Analysis of Curved Line Drawings Using Context and Global Information", in Machine Intelligence, B. Meltzer and D. Michie (eds), American Elsevier Publishing Company, New York, New York, v 6, pp 325-375.

Harris, Z. (1968). Mathematical Structures of Language, Interscience Publishers, New York, New York.

Hempel, C. (1939). "Vagueness and Logic", Philosophy of Science, v 6, pp 163-180.

Hendrix, G. G., Thompson, C. W., and Slocum, J. (1973). "Language Processing via Canonical Verbs and Semantic Models", Advance Papers of Third International Joint Conference on Artificial Intelligence, Stanford, California, pp 262-269.

Heny, F. W. (1973). "Sentence and Predicate Modifiers in English", in Syntax and Semantics, vol. 2, J. Kimball (ed), Seminar Press, New York, New York, pp 217-245.

Hewitt, C. (1971). "Description and Theoretical Analysis of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot", PhD Thesis, MIT, Cambridge, Massachusetts

Howe, W. and Krulee, G. (1971). "A Logic of English Questions", IBM Research, RC 3490, Yorktown Heights, New York.

Isard, S. and Longuet-Higgins, H. C. (1971). "Question-Answering in English", in Machine Intelligence, B. Meltzer and D. Michie (eds), American Elsevier Publishing Company, New York, New York, v 6, pp 243-254.

Johnson-Laird, P. (1969). "On Understanding Logically Complex Sentences", Quarterly Journal of Experimental Psychology, v 21, pp 1-13.

Jordan, S. (1972). "Learning to Use Contextual Patterns in Language Processing", University of Wisconsin, Technical Report No. 152, Madison, Wisconsin.

Kalish, D. and Montague, R. (1964). Logic: Techniques of Formal Reasoning, Harcourt, Brace, and World, New York, New York.

Kaplan, R. (1972). "Augmented Transition Networks as Psychological Models of Sentence Comprehension", Artificial Intelligence Journal, v 3, n 2, pp 77-100.

Katz, J. and Fodor, J. (1964). "The Structure of Semantic Theory", in The Structure of Language, J. Fodor and J. Katz (eds), Prentice-Hall, Englewood Cliffs, New Jersey.

Kripke, S. (1972). "Naming and Necessity", in Semantics of Natural Language, D. Davidson and G. Harman (eds), D. Reidel Publishing Company, Boston, Massachusetts

Kuno, S. (1965). "The Predictive Analyzer and a Path Elimination Technique", Communications of the Association for Computing Machinery, v 8, n 7, pp 453-462.

Lakoff, G. (1972). "Linguistics and Natural Logic", in Semantics of Natural Language, D. Davidson and G. Harman (eds), D. Reidel Publishing Company, Boston, Massachusetts

Lakoff, G. (1973). "Hedges: A Study in Meaning Criteria and the Logic of Fuzzy Concepts", Journal of Philosophical Logic, v 2, pp 458-508.

Levi, I. (1973). Gambling With Truth, MIT Press, Cambridge, Massachusetts.

Lewis, D. (1972). "General Semantics", in Semantics of Natural Language, D. Davidson and G. Harman (eds), D. Reidel Publishing Company, Boston, Massachusetts

Lewis, D. (1973). Counterfactuals, Harvard University Press, Cambridge, Massachusetts.

Lindsay, R. (1963). "Inferential Memory as the Basis of Machines Which Understand Natural Thought", in Computers and Thought, E. Feigenbaum and J. Feldman (eds), McGraw Hill Book Company, New York, New York, pp 217-233.

Lyons, J. (1968). Introduction to Theoretical Linguistics, Cambridge University Press, Cambridge, England.

McCalla, G. and Sampson, J. (1972). "MUSE: A Model to Understand Simple English", Communications of the Association fcr Computing Machinery, v 15, n 1, pp 29-40.

McCarthy, J. and Hayes, P. (1969). "Some Philosophical Problems from the Standpoint of Artificial Intelligence", in Machine Intelligence, B. Meltzer and D. Michie (eds), American Elsevier Publishing Company, New York, New York.

McConlogue, K. and Simmons, R. (1965). "Analyzing English Syntax with a Pattern Learning Parser", Communications of the Association fcr Computing Machinery, v 8, n 11, pp 687-698.

McDermott, D. (1974). "Assimilation of New Information by a Natural Language Understanding System", AI TR-291, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetes.

McDermott, D., and Sussman, G. (1972). "The Conniver Reference Manual", Memo No. 259, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetes.

Minsky, M. (1974). "A Framework for Representing Knowledge", Artificial Intelligence Memo No. 306, A. I. Laboratory, MIT, Cambridge, Massachusetts.

Montague, R. (1969). "English as a Formal Language", reprinted in Formal Philosophy: Selected Papers of Richard Montague, R. Thomason (ed), Yale University Press, New Haven, Connecticut, 1974, pp 188-221.

Montague, R. (1970). "The Proper Treatment of Quantification in Ordinary English", Department of Philosophy, University of California, Los Angeles, California.

Montague, R. (1972). "Pragmatics and Intensional Logic", in Semantics of Natural Language, D. Davidson and G. Harman (eds), D. Reidel Publishing Company, Boston, Massachusetts

Moore, R. (1973). "D-SCRIPT: A Computational Theory of Descriptions", Memo No. 278, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts

Moore, J. and Newell, A. (1973). "How Can Merlin Understand", Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

Nagel, E. (1939). "Probability and the Theory of Knowledge", Philoscphy of Science, v 6, pp 212-253.

Nilsson, N. (1971). Problem Solving Methods in Artificial Intelligence, McGraw Hill Bock Company, New York, New York.

Palme, J. (1971). "Making Computers Understand Natural Language", in Artificial Intelligence and Heuristic Programming, N. Findler and B. Meltzer (eds), American Elsevier Publishing Company, New York, New York, pp 199-244.

Partee, B. (1972). "Opacity, Coreference, and Pronouns", in Semantics of Natural Language, D. Davidson and G. Harman (eds), D. Reidel Publishing Company, Boston, Massachusetts

Phillips, A. (1960). "A Question-Answering Routine", AI Memo No. 16, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts.

Piaget, J. and Beth, E. (1966). Mathematical Epistemology and Psychclcgy, D. Reidel Publishing Company, Holland.

Quillian, M. (1968). "Semantic Memory", in Semantic Information Processing, M. Minsky (ed), MIT Press, Cambridge, Massachusetts, pp 227-270.

Quillian, M. (1969). "The Teachable Language Comprehender", Communications of the Association fcr Computing Machinery, v 12, n 8, pp 459-475.

Quine, W., (1960). Word and Object, MIT Press, Cambridge, Massachusetts.

Quine, W., (1961). From a Logical Pcint of View, Harper Torchbooks, New York, New York.

Raphael, B. (1968). "SIR: A Computer Program for Semantic Information Retrieval", in Semantic Information Processing, M. Minsky (ed), MIT Press, Cambridge, Massachusetts, pp 33-145.

Reichenbach, H. (1947). Elements of Symbolic Logic, paperback editicn, New York Free Press, New York, New York, 1966.

Reitman, W. (1965). Cognition and Thought, John Wiley and Sons, New York, New York.

Rosser, J., and Turquette, A. (1952). Many-valued Logics, Amsterdam, North Holland.

Rulifson, J., Derksen, J., and Waldinger, R. (1972). "QA4: A Procedural Calculus for Intuitive Reasoning", AI Center, Technical Note 73, Stanford Research Institute, Menlo Park, California.

Rumelhart, D., Lindsay, P., and Norman, D. (1972). "A Process Model for Long Term Memory", in Organization of Memory, E. Tulving and W. Donaldson (eds), Academic Press, New York, New York, pp 198-221.

Russell, B. (1923). "Vagueness", Australasian Journal of Philosophy, v 1, pp 84-92.

Russell, S. (1972). "Semantic Categories of Nominals for Conceptual Dependency Analysis of Natural Language", Stanford AI Project, Memo AIM-172, Stanford University, Stanford, California.

Sandewall, E. J. (1970). "A Set-Oriented Property-Structure Representation for Binary Relations", in Machine Intelligence, B. Meltzer and D. Michie (eds), American Elsevier Publishing Company, New York, New York, v 5, pp 237-252.

Sandewall, E. J. (1971a). "Representing Natural Language Information in Predicate Calculus", in Machine Intelligence, B. Meltzer and D. Michie (eds), American Elsevier Publishing Company, New York, New York, v 6, pp 255-277.

Sandewall, E. (1971b). "Formal Methods in the Design of Question-Answering Systems", Artificial Intelligence Journal, v 2, n 2, pp 129-145.

Schank, R. (1972). "Conceptual Dependency: A Theory of Natural Language Understanding", Cognitive Psychology, v 3, pp 552-631.

Schank, R. (1973a). "The Fourteen Primitive Actions and Their Inferences", Stanford AI Project, Memo AIM-183, Stanford University, Stanford, California.

Schank, R. (1973b). "Identification of Conceptualizations Underlying Natural Language", in Computer Models of Thought and Language, R. Schank and K. Colby (eds), W. H. Freeman and Company, San Francisco, California.

Schank, R. (1974). "Adverbs of Belief", LINGUA, v 33, North Holland Publishing Company, pp 45-67.

Schank, R., (1975). "The Role of Memory in Language Processing", Yale University Technical Report, New Haven, Connecticut.

Schank, R., Goldman, N., Rieger, C., and Riesbeck, C. (1973). "Margie: Memory, Analysis, Response Generation, and Inference on English", Third International Joint Conference on Artificial Intelligence, Stanford, California, pp 255-261.

Schubert, L. (1974). "On the Expressive Adequacy of Semantic Networks", TR74-18, Department of Computing Science, University of Alberta, Edmonton, Alberta.

Schwarcz, R. (1967). "Steps Toward a Model of Linguistic Performance: A Preliminary Sketch", Mechanical Translation, v 10, pp 39-52.

Scragg, W. (1974). PhD Thesis, University of California, San Diego, San Diego, California.

Shapiro, S. and Woodmansee, G. (1969). "A Net Structure Based Relational Question-Answerer: Description and Examples", Proceedings of the International Joint Conference on Artificial Intelligence, Bedford, Massachusetts, pp 325-346.

Siklossy, L. (1968). "Natural Language Learning by Computer", PhD Thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

Simmons, R. (1968). "Answering English Questions by Computer", in Automated Language Processing, H. Borko (ed), John Wiley and Sons, New York, New York, pp 253-290.

Simmons, R. (1970). "Natural Language Question-Answering Systems: 1969", Communications of the Association for Computing Machinery, v 13, n 1, pp 15-30.

Simmons, R. F. (1972). " ", in Language Comprehension and the Acquisition of Knowledge, J. Carroll and R. Freedle (eds), V. H. Winston, Washington, D.C.

Simmons, R. (1973). "Semantic Networks: Their Computation and Use for Understanding English", in Computer Models of Thought and Language, R. Schank and K. Colby (eds), W. H. Freeman, San Francisco, California.

Simmons, R., Klein, S., and McConlogue, K. (1964). "Indexing and Dependency Logic for Answering English Questions", American Documentation, v 15, n 3, pp 196-204.

Simmons, R., Burger, R., and Long, R. (1966). "An Approach Toward Answering English Questions from Text", Proceedings of the Fall Joint Computer Conference, Spartan, New York, pp 357-363.

Simmons, R., Burger, R., and Scwartz, R. (1968). "A Computerized Model of Verbal Understanding", Proceedings of the Fall Joint Computer Conference, Spartan, New York, pp 441-456.

Simmons, R. F., and Bruce, B. C. (1971). "Some Relations Between Predicate Calculus and Semantic Net Representation of Discourse", Advance Papers of Second International Joint Conference on Artificial Intelligence, The British Computer Society, London, England, pp 524-530.

Simmons, R. F. and Slocum, J. (1972). "Generating English Discourse from Semantic Networks", <u>Communications of the Association for Computing Machinery</u>, v 15, pp 891-905.

Slagle, J. (1965). "Experiments with a Deductive Question-Answering Program", <u>Communications of the Association for Computing Machinery</u>, v 8 ,n 12, pp 792-798.

Slagle, J. (1971). <u>Artificial Intelligence: The Heuristic Programming Approach</u>, McGraw Hill Book Company, New York, New York.

Sloman, A. (1971). "Interactions Between Philosophy and Artificial Intelligence: The Role of Intuition and Nonlogical Reasoning in Intelligence", Advance Papers of Second International Joint Conference on Artificial Intelligence, The British Computer Society, London, England, pp 270-279.

Snyder, D. P. (1971). <u>Modal Logic and its Applications</u>, Van Nostrand Reinhold, New York, New York.

Stalnaker, R., (1972). "Pragmatics", in <u>Semantics of Natural Language</u>, D. Davidson and G. Harman (eds), D. Reidel Publishing Company, Boston, Massachusetts

Sussman, G., et al. (1972). "Microplanner Reference Mannual", Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetes.

Sussman, G. (1972). "Why Conniving is Better than Planning", Memo No. 255A, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetes.

Svensson, B. (1974). "On the Use of Fuzzy Concepts in Description", Proceedings of the Second International Joint Conference on Pattern Recognition, Copenhagen, Denmark, pp 162-168.

Tharp, A. and Krulee, G. (1969). "Using Relational Operators to Structure Long Term Memory", Proceedings of the International Joint Conference on Artificial Intelligence, Bedford, Massachusetts, pp 579-587.

Thompson, F. et al. (1964). "DEACON breadboard summary", RM64TMP-9, General Electric, Santa Barbara, California.

Thorne, J., Bratley, P., and Dewar, H. (1968). "The Syntactic Analysis of English by Machines", in Machine Intelligence, B. Meltzer and D. Michie (eds), American Elsevier Publishing Company, New York, New York, v 3, pp 281-309.

Tulving, E. and Donaldson, W. (1972). Organization of Memory, Academic Press, New York, New York.

Turing, A. (1950). "Computing Machinery and Intelligence", Mind, v 59, pp 433-460.

Vigor, D., Urquhart, D., and Wilkinson, A. (1969). "PROSE: Parsing Recognizer Outputting Sentences in English", in Machine Intelligence, B. Meltzer and D. Michie (eds), American Elsevier Publishing Company, v 4, New York, New York, pp 271-284.

Weinreich, U. (1963). "On the Semantic Structure of Language", in Universals of Language, J. Greenberg (ed), MIT Press, Cambridge, Massachesetts.

Weizenbaum, J. (1966). "ELIZA", Communications of the Association for Computing Machinery, v 9, n 1, pp 36-45.

Weizenbaum, J. (1967). "Contextual Understanding by Computers", Communications of the Association for Computing Machinery, v 10, n 8, pp 474-480.

Wheatley, J. (1970). Language and Rules, Mouton, The Hague.

Wilks, Y. (1973a). "Preference Semantics", Stanford AI Project, Memo AIM-206, Stanford University, Stanford, California.

Wilks, Y. (1973b). "Natural Language Inference", Stanford AI Project, Memo AIM-211, Stanford University, Stanford, California.

Wilks, Y. (1973c). "Understanding Without Proofs", Third International Joint Conference on Artificial Intelligence, Stanford, California, pp 270-277.

Winograd, T. (1969). "PROGRAMMAR: A Language for Writing Grammars", Memo No. 181, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts

Winograd, T. (1971). "An Approach to English Morphemic Analysis", Memo No. 241, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts

Winograd, T. (1972). Understanding Natural Language, Academic Press, New York, New York.

Winograd, T. (1974). "Five Lectures on Artificial Intelligence", Stanford AI Project, Memo AIM-246, Stanford University, Stanford, California.

Winston, P. (1970). "Learning Structural Descriptions from Examples", PhD Thesis, MIT, MAC-TR-76, Cambridge, Massachusetts

Woods, W. (1970). "Transition Networks for Natural Language Analysis", Communications of the Association for Computing Machinery, v 13, n 10, pp 591-606.

Woods, W., Kaplan, R., and Nash-Webber, B. (1972). "The Lunar Sciences Natural Language Information System: Final Report", Bolt Benarek and Newman Inc., Cambridge, Massachusetts.

Zadeh, L. (1965). "Fuzzy Sets", Information and Control, v 8, pp 338-353.

Zadeh, L. (1972). "A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges", Journal of Cybernetics, v 2, n 3, pp 4-34.

Appendix A Uses of English Morphology

Morphological analysis is an integral part of natural language understanding systems. There are many and varied reasons for this. The most important reasons include the following.

(i) Storage economy

> It would be absurd to store all forms of lexical items directly since well-defined spelling rules exist that specify all normal word formations. The use of a small, relatively simple analysis routine can save vast amounts of storage that otherwise would be needed to save all forms of a lexical item.

(ii) Interpretive assistance

> A by-product of morphological analysis is the discovery of affixes that were added to the root word to form the word under analysis. Often these affixes, especially in the case of inflexional endings, determine the use of the word in an utterance, or, at least narrow its possibilities. For example a word ending in "er" can only be a comparative or a personal noun.

(iii) Learning new words

> Whenever an unknown word is encountered in text, preliminary analysis using morphological information can be used to aid in determining the word's function in the utterance. Our ability to infer or guess meanings of unfamiliar words can only be enhanced through a more extensive morphological analysis.

(iv) Derivational information

> Derivational affixes affect word meaning, often in a systematic way, for example non-, negation; -esque, like a. More specifically, a word like "booklet" can be satisfactorily understood as "little book" through the use of morphological analysis without explicitly storing both book and booklet.

The morphological component of the experimental program has been described elsewhere, see Cercone (1974). Since the

time of that report improvements have been made to handle the analyses of additional derivational affixes and to speed up processing.

The following is a sample output from the morphological component working with a lexical structure as described in Chapter 5. It was written in a converted MACLISP modified to run on an IBM 360/67 under the Michigan Timesharing System [MTS]. The STEM routine returns a list containing the original word analyzed, any possible prefixes, the root form of the word, and a list of suffixes. In this way no information has been lost.

```
# R NEW:MACLISP
# 21:50.18
=                    (RESTORE 'CHKPT)
=   NIL
=                                    (STEM 'INTEREST)
=   (INTEREST NIL INTEREST NIL)
=                                    (STEM 'INTERESTED)
=   (INTERESTED NIL INTEREST (ED))
=                                    (STEM 'INTERESTING)
=   (INTERESTING NIL INTEREST (ING))
=                                    (STEM 'INTERESTINGLY)
=   (INTERESTINGLY NIL INTEREST (ING LY))
=                                    (STEM 'RUN)
=   (RUN NIL RUN NIL)
=                                    (STEM 'RUNS)
=   (RUNS NIL RUN (S))
=                                    (STEM 'RUNNING)
=   (RUNNING NIL RUN (ING))
=                                    (STEM 'RUNNER)
=   (RUNNER NIL RUN (ER))
=                                    (STEM 'RUNNERS)
=   (RUNNERS NIL RUN (ER S))
=                                    (STEM 'RUNNER"S)
=   (RUNNER/"S NIL RUN (ER /"S))
=                                    (STEM 'RUNNERS")
=   (RUNNERS/" NIL RUN (ER S /"))
=                                    (STEM 'BASHES)
=   (BASHES NIL BASH (S))
=
```

```
=                                      (STEM 'BATHES)
=    (BATHES NIL BATHE (S))
=                                      (STEM 'BATHS)
=    (BATHS NIL BATH (S))
=                                      (STEM 'LEANING)
=    (LEANING NIL LEAN (ING))
=                                      (STEM 'LEAVING)
=    (LEAVING NIL LEAVE (ING))
=                                      (STEM 'DENTED)
=    (DENTED NIL DENT (ED))
=                                      (STEM 'DANCED)
=    (DANCED NIL DANCE (ED))
=                                      (STEM 'KISSES)
=    (KISSES NIL KISS (S))
=                                      (STEM 'CURVED)
=    (CURVED NIL CURVE (ED))
=                                      (STEM 'CURLED)
=    (CURLED NIL CURL (ED))
=                                      (STEM 'ROTTING)
=    (ROTTING NIL ROT (ING))
=                                      (STEM 'ROLLING)
=    (ROLLING NIL ROLL (ING))
=                                      (STEM 'PLAYED)
=    (PLAYED NIL PLAY (ED))
=                                      (STEM 'PLIED)
=    (PLIED NIL PLY (ED))
=                                      (STEM 'REALEST)
=    (REALEST NIL REAL (EST))
=                                      (STEM 'PALEST)
=    (PALEST NIL PALE (EST))
=                                      (STEM 'KNIVES)
=    (KNIVES NIL KNIFE (S))
=                                      (STEM 'PRETTILY)
=    (PRETTILY NIL PRETTY (LY))
=                                      (STEM 'NOBLY)
=    (NOBLY NIL NOBLE (LY))
=                                      (STEM 'PATROLLING)
=    (PATROLLING NIL PATROL (ING))
=                                      (STEM 'PATRONIZINGLY)
=    (PATRONIZINGLY NIL PATRON (IZE ING LY))
=                                      (STEM 'RELIABLE)
=    (RELIABLE NIL RELY (ABLE))
=                                      (STEM 'BREAKABLE)
=    (BREAKABLE NIL BREAK (ABLE))
=                                      (STEM 'ACCESSIBLE)
=    (ACCESSIBLE NIL ACCESS (IBLE))
=                                      (STEM 'ACREAGE)
=    (ACREAGE NIL ACRE (AGE))
=                                      (STEM 'MILAGE)
=    (MILAGE NIL MILE (AGE))
=                                      (STEM 'STOPPAGE)
=    (STOPPAGE NIL STOP (AGE))
=
```

```
=                                          (STEM 'CULTURAL)
=    (CULTURAL NIL CULTURE (AL))
=                                          (STEM 'MUSICAL)
=    (MUSICAL NIL MUSIC (AL))
=                                          (STEM 'RIDDANCE)
=    (RIDDANCE NIL RID (ANCE))
=                                          (STEM 'UTTERANCE)
=    (UTTERANCE NIL UTTER (ANCE))
=                                          (STEM 'DISTANCE)
=    (DISTANCE NIL DISTANCE NIL)
=                                          (STEM 'OPERATION)
=    (OPERATION NIL OPERATE (ATION))
=                                          (STEM 'ACCURACY)
=    (ACCURACY NIL ACCURATE (CY))
=                                          (STEM 'CONSTANCY)
=    (CONSTANCY NIL CONSTANT (CY))
=                                          (STEM 'CAPTAINCY)
=    (CAPTAINCY NIL CAPTAIN (CY))
=                                          (STEM 'WISDOM)
=    (WISDOM NIL WISE (DOM))
=                                          (STEM 'PATRIOTISM)
=    (PATRIOTISM NIL PATRIOT (ISM))
=                                          (STEM 'SOCIALIST)
=    (SOCIALIST NIL SOCIAL (IST))
=                                          (STEM 'VISIBILITY)
=    (VISIBILITY NIL VISIBLE (ITY))
=                                          (STEM 'COMICALITY)
=    (COMICALITY NIL COMIC (AL ITY))
=                                          (STEM 'SENTIMENTALITY)
=    (SENTIMENTALITY NIL SENTIMENT (AL ITY))
=                                          (STEM 'CIVILIZE)
=    (CIVILIZE NIL CIVIL (IZE))
=                                          (STEM 'PENNILESS)
=    (PENNILESS NIL PENNY (LESS))
=                                          (STEM 'RESTLESS)
=    (RESTLESS NIL REST (LESS))
=                                          (STEM 'CHILDLIKE)
=    (CHILDLIKE NIL CHILD (LIKE))
=                                          (STEM 'ARGUMENT)
=    (ARGUMENT NIL ARGUE (MENT))
=                                          (STEM 'EMBODIMENT)
=    (EMBODIMENT NIL EMBODY (MENT))
=                                          (STEM 'TREATMENT)
=    (TREATMENT NIL TREAT (MENT))
=                                          (STEM 'DRUNKENNESS)
=    (DRUNKENNESS NIL DRUNK (EN NESS))
=                                          (STEM 'WICKEDNESS)
=    (WICKEDNESS NIL WICKED (NESS))
=                                          (STEM 'MOUNTAINOUS)
=    (MOUNTAINOUS NIL MOUNTAIN (OUS))
=                                          (STEM 'NERVOUS)
=    (NERVOUS NIL NERVE (OUS))
=
```

```
=                                          (STEM 'FRIENDSHIPS)
=    (FRIENDSHIPS NII FRIEND (SHIP S))
=                                          (STEM 'WEARISOME)
=    (WEARISOME NIL WEARY (SOME))
=                                          (STEM 'FEARSOME)
=    (FEARSOME NIL FEAR (SOME))
=                                          (STEM 'GANGSTER)
=    (GANGSTER NIL GANG (STER))
=                                          (STEM 'HOMEWARD)
=    (HOMEWARD NIL HOME (WARD))
=                                          (STEM 'ASLEEP)
=    (ASLEEP A SLEEP NIL)
=                                          (STEM 'ANTEROOM)
=    (ANTEROOM ANTE ROOM NIL)
=                                          (STEM 'ANTICHRIST)
=    (ANTICHRIST ANTI CHRIST NIL)
=                                          (STEM 'ARCHBISHOP)
=    (ARCHBISHOP ARCH BISHOP NIL)
=                                          (STEM 'AUTOBIOGRAPHY)
=    (AUTOBIOGRAPHY AUTO BIOGRAPHY NIL)
=                                          (STEM 'BEMOAN)
=    (BEMOAN BE MOAN NIL)
=                                          (STEM 'BIANNUAL)
=    (BIANNUAL BI ANNUAL NIL)
=                                          (STEM 'COHEIR)
=    (COHEIR CO HEIR NIL)
=                                          (STEM 'DECODE)
=    (DECODE DE CODE NIL)
=                                          (STEM 'DISTRUST)
=    (DISTRUST DIS TRUST NIL)
=                                          (STEM 'ENDANGER)
=    (ENDANGER EN DANGER NIL)
=                                          (STEM 'EMBED)
=    (EMBED NIL EMBED NIL)
=                                          (STEM 'INTERRELATION)
=    (INTERRELATION INTER RELATE (ATION))
=                                          (STEM 'INTERRELATIONSHIP)
=    (INTERRELATIONSHIP INTER RELATE (ATION SHIP))
=                                          (STEM 'MALPRACTICE)
=    (MALPRACTICE MAL PRACTICE NIL)
=                                          (STEM 'MISCONDUCT)
=    (MISCONDUCT MIS CONDUCT NIL)
=                                          (STEM 'NONSTOP)
=    (NONSTOP NON STOP NIL)
=                                          (STEM 'POSTWAR)
=    (POSTWAR POST WAR NIL)
=                                          (STEM 'PREARRANGE)
=    (PREARRANGE PRE ARRANGE NIL)
=                                          (STEM 'PREWAR)
=    (PREWAR PRE WAR NIL)
=                                          (STEM 'RECONSIDER)
=    (RECONSIDER RE CONSIDER NIL)
=
```

```
=                                                 (STEM 'REFUEL)
=    (REFUEL RE FUEL NIL)
=                                                 (STEM 'UNNECESSARY)
=    (UNNECESSARY UN NECESSARY NIL)
=                                                 (STEM 'UNREST)
=    (UNREST UN REST NIL)
=                                                 (STEM 'COEDUCATION)
=    (COEDUCATION CO EDUCATE (ATION))
=                                                 (STEM 'COEXISTINGLY)
=    (COEXISTINGLY CO EXIST (ING LY))
=                                                 (STEM 'COOPERATIONAL)
=    (COOPERATIONAL CO OPERATE (ATION AL))
=                                                 (STEM 'DEHUMANIZE)
=    (DEHUMANIZE DE HUMAN (IZE))
=                                                 (STEM 'INEQUALITY)
=    (INEQUALITY IN EQUAL (ITY))
=                                                 (STEM 'MALODOROUS)
=    (MALODOROUS MAL ODOR (OUS))
=                                                 (STEM 'NONPAYMENT)
=    (NONPAYMENT NON PAY (MENT))
=                                                 (STEM 'REELIGIBILITY)
=    (REELIGIBILITY RE ELIGIBLE (ITY))
=                                                 (STEM 'LOUDLY)
=    (LOUDLY NIL LOUD (LY))
=                                                 (STEM 'LENGTHWISE)
=    (LENGTHWISE NIL LENGTH (WISE))
=                                                 (STEM 'LENGTHWAYS)
=    (LENGTHWAYS NIL LENGTH (WAY S))
=                                                 (STEM 'NOWAYS)
=    (NOWAYS NIL NO (WAY S))
=                                                 (STEM 'UPWARD)
=    (UPWARD NIL UP (WARD))
=                                                 (MTS)
```

The first example in Appendix C shows a further use of morphological analysis. In particular it shows how morphological analysis can help to narrow a word's function (and in part its sense) in an utterance.

Appendix B Sample Lexical Entries and Lexical Maintainence

In order to enable the rapid retrieval of lexical information regardless of the lexical size, the following scheme was develcped and structure imposed on the lexicon. The form of the root of a lexical item is that of a binary branching tree that suggests a hash similar to a binary search. Letters in the query word "hash" to a subset of lexical entries that contain the letters in corresponding positions. For example, a word like "drink" would be found as follows. The letter "d" would hash to the lexical items beginning with "d". All other lexical items would not be considered further. The letter "r" would hash to all lexical items that begin with "dr" and so on until the word "drink" is found. In this way the number of probes of the lexicon needed to locate a lexical item is directly proportional to the number of letters in the word. Each letter in a query word causes us to eliminate all but words that have that letter in that position. This is easily done in LISP.

The following are sample lexical entries, first from the closed category (Figure B1) and then form the open category (Figure B2) lexicon.[23]

---

[23] In actual implementation there is only one lexicon and dictionary routines exist for combining (merging) several dictionaries, inserting new entries in dictionaries, or even for creating new dictionaries.

```
(B (E (F (O (R (E (* (BIND () (*BEFORE1))
                     (PREP () (*BEFORE2))
                     ((AM (AIT)) (*BEFORE3))) ))))
    (H (I (N (D (* (PREP () (*BEHIND))) ))))
    (L (O (W (* (PREP () (*BELOW1))
                (AM ((AP AA)) (*BELOW2))) )))
    (N (E (A (T (H (* (PREP () (*BENEATH1))
                      (AM ((AP AA)) (*BENEATH2))) )))))
    (S (I (D (E (* (PREP () (*BESIDE))) )))))
  (O (T (H (* (QNTRF ((NP COLL)) (*BOTH1))
              (PRO ((INDEF)) (*BOTH2)) (AM () (*BOTH3))) )))
  (U (T (* (BIND () (*BUT1)) (AM () (*BUT2))) ))
  (Y (* (PREP () (*BY1)) (PRT () (*BY2))) ))
(D (O (W (N (* (PREP () (*DOWN1)) (PRT () (*DOWN2))) ))))
(E (A (C (H (* (QNTRF ((NS)) (*EACH1)))))
              (PRO ((INDEF NS COLL)) (*EACH2))) )))
  (I (T (H (E (R (* (QNTRF ((NS NP)) (*EITHER1))
                    (PRO ((INDEF NS NP)) (*EITHER2))) ))))
    (G (H (T (* (QNTRF ((NP)) (*EIGHT1)) (NUM () (*EIGHT2)))
             (H (* (ORD () (*EIGHTH))) )))))
  (L (S (E (* (AM () (*ELSE))) )))
  (V (E (R (Y (* (QNTRF ((NS)) (*EVERY)))
             (O (N (E (* (PRO ((INDEF)) (*EVERYONE))) )))
             (T (H (I (N (G (* (PRO ((INDEF NS)) (*EVERYTHING)))))))))))
  (X (C (E (P (T (* (PREP () (*EXCEPT)) (CONJ () (*EXCEPT2))) ))))))
(F (E (W (* (QNTRF ((NONUM NP COLL)) (*FEW)))
        (E (R (* (QNTRF ((NONUM NP COLL)) (*FEW))) ))))
  (I (F (T (H (* (ORD () (*FIFTH))) )))
    (R (S (T (* (ORD () (*FIRST))) )))
    (V (E (* (QNTFR ((NP)) (*FIVE1)) (NUM () (*FIVE2))))))
  (O (R (* (PREP () (*FOR1)) (CONJ () (*FOR2))) )
    (U (R (* (QNTRF ((NP)) (*FOUR1)) (NUM () (*FOUR2)))
          (T (H (* (ORD () (*FOURTH))) )))))
  (R (O (M (* (PREP () (*FROM))) ))))
```

Fig. B.1. Closed Category Lexical Entries

```
(D(R(A(N(K(*(A  ((NIL IREG PAST))
                 ((( 0 0) (*DRINK1 P1 P2))
                  ((0 0) (*DRINK1A P1 P2))
                  ((0 0) (*DRINK3 P1 P2))) ) ))))
    (I(N(K(*(N  ((NIL NS)(S NP)(ABLE NS)(ETTE DIM)(IE DIM))
                 (((0 0) (*DRINK2))
                  ((0 0) (*DRINK4)))
                 ((ING NS))
                 (((0 0) (*DRINK6)))
                 ((ER PERS)(EER PERS)(IST PERS))
                 (((0 0) (*DRINK5)))
                 (SYN DRAFT POTATION BEVERAGE LIQUOR)
                 (ID BOOZE HOOCH MOONSHINE) )
            (A  ((NIL PRES)(S PRES TPS)(ING PART))
                 (((0 0) (*DRINK1 P1 P2))
                  ((0 0) (*DRINK1A P1 P2))
                  ((0 0) (*DRINK3 P1 P2)))
                 (SYN CONSUME SWALLOW IMBIBE GUZZLE TOAST)
                 (ID SWIG SOP-UP) )
            (AM  ((WISE AT3)(WAYS AT3))
                  (((0 0) (*DRINK1 KIND))
                   ((0 0) (*DRINK1A KIND))) )
            (NM  ((ABLE ADJ CLASF)(ING ADJ CLASF)(LIKE ADJ))
                  (((0 0) (*DRINK2 P1 P2))
                   ((0 0) (*DRINK4 P1 P2))
                   ((0 0) (*DRINK5 P1 P2))
                   ((0 0) (*DRINK6 P1 P2))) ) ))))
    (U(N(K(*(N  ((NIL NS)(S NP)) (((0 0) (*DRUNK1))) )
            (A  ((NIL IREG PART))
                 (((0 0) (*DRINK1 P1 P2))
                  ((0 0) (*DRINK1A P1 P2))
                  ((0 0) (*DRINK3 P1 P2))) ) ))))))
 (E(A(T(*(N  ((S NP)(IE DIM))
             (((0 0) (*EAT3)))
             ((ER PERS)(EER PERS))
             (((0 0) (*EAT3)))
             (SYN FOOD)
             (ID MUNCHIES GRUB FULLERS GRUMBLIES) )
         (A  ((NIL PRES)(S PRES TPS)(ING PART))
             (((0 0) (*EAT1 P1 P2))
              ((0 0) (*EAT2 P1 P2)))
             (SYN CONSUME DEVOUR FEED FARE ERODE WEAR)
             (ID GOBBLE) )
         (NM  ((ABLE ADJ CLASF)(ING ADJ CLASF))
              (((0 0) (*EAT3))) ) )
      (E(N(*  (A  ((NIL IREG PART))
                  (((0 0) (*EAT1 P1 P2))
                   ((0 0) (*EAT2 P1 P2))) ) )))))
```

Fig. B.2. Open Category Lexical Entries

The following example shows (in order) a search through dictionaries for lexical items, the merging of two dictionaries, a search for the same lexical items just found in the merged version, an addition to the merged dictionary, and finally a search for the newly added lexical item.

```
# R NEW:MACLISP
# 16:07.28
= (RESTORE 'CHKPT1)
=      NIL
= (VERBOS NIL)
=      NIL
= (OTLL 60)
=      NIL
=
= (CHK '(S O M E) PRED)
=      NIL
=
= (CHK '(D R I N K) PRED)
=      (((N ((NIL NS) (S NP) (ABLE NS) (ETTE DIM) (IE
=      DIM)) (((0 0) (/*DRINK1 P2)) ((0 0) (/*DRINK1A
=      P2)) ((0 0) (/*DRINK2 P1 P2)) ((0 0) (/*DRINK3
=      P2 P11))) ((ING NS)) (((0 0) (/*DRINK1A ACT DO)
=      )) ((ER PERS) (EER PERS) (IST PERS)) (((0 0) (/*DRINK1
=      P1)) ((0 0) (/*DRINK1A P1)) ((0 0) (/*DRINK3 P1)
=      )) (SYN DRAFT POTATION BEVERAGE LIQUOR) (ID BOOZE
=      HOOCH MOONSHINE)) (A ((NIL PRES) (S PRES TPS) (
=      ING PART)) (((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A
=      P1 P2)) ((0 0) (/*DRINK3 P1 P2))) (SYN CONSUME
=      SWALLOW IMBIBE GUZZLE TOAST) (ID SWIG SOP/-UP))
=       (AM ((WISE AT3) (WAYS AT3)) (((0 0) (/*DRINK1
=      KIND)) ((0 0) (/*DRINK1A KIND)))) (NM ((ABLE ADJ
=      CLASF) (ING ADJ CLASF) (LIKE ADJ)) (((0 0) (/*DRINK1
=      P2)) ((0 0) (/*DRINK1A P2)) ((0 0) (/*DRINK3 P2)
=      )))))
=
= (CHK '(S E X U A L) PRED)
=      NIL
=
= (CHK '(S O M E) ADVERB)
=      NIL
=
= (CHK '(D R I N K) ADVERB)
=      NIL
=
= (CHK '(S E X U A L) ADVERB)
=      (((AM ((LY AT3 AAA AA)) (/*SEXUAL1))))
=
```

```
=
= (COMBINE PRED ADVERB '* NIL)
=     NIL
=
= (CHK '(S E X U A L) PRED)
=     (((AM ((LY AT3 AAA AA)) (/*SEXUAL1))))
=
= (CHK '(S O M E) PRED)
=     NIL
=
= (CHK '(D R I N K) PRED)
=     (((N ((NIL NS) (S NP) (ABLE NS) (ETTE DIM) (IE
=     DIM)) (((0 0) (/*DRINK1 P2)) ((0 0) (/*DRINK1A
=     P2)) ((0 0) (/*DRINK2 P1 P2)) ((0 0) (/*DRINK3
=     P2 P11))) ((ING NS)) (((0 0) (/*DRINK1A ACT DO)
=     )) ((ER PERS) (EER PERS) (IST PERS)) (((0 0) (/*DRINK1
=     P1)) ((0 0) (/*DRINK1A P1)) ((0 0) (/*DRINK3 P1)
=     )) (SYN DRAFT POTATION BEVERAGE LIQUOR) (ID BOOZE
=     HOOCH MOONSHINE)) (A ((NIL PRES) (S PRES TPS) (
=     ING PART)) (((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A
=     P1 P2)) ((0 0) (/*DRINK3 P1 P2))) (SYN CONSUME
=     SWALLOW IMBIBE GUZZLE TOAST) (ID SWIG SOP/-UP))
=      (AM ((WISE AT3) (WAYS AT3)) (((0 0) (/*DRINK1
=     KIND)) ((0 0) (/*DRINK1A KIND)))) (NM ((ABLE ADJ
=     CLASF) (ING ADJ CLASF) (LIKE ADJ)) (((0 0) (/*DRINK1
=     P2)) ((0 0) (/*DRINK1A P2)) ((0 0) (/*DRINK3 P2)
=     )))))
=
=
= (DICTADD PRED '(S O M E) '*
=   '((PRO (INDEF NS NP)) (QNTRF (COLL NS NP NONUM))) )
=     NIL
=
= (CHK '(S O M E) PRED)
=     (((PRO (INDEF NS NP)) (QNTRF (COLL NS NP NONUM)
=     )))
=
= (MTS)
```

Appendix C Sample Text Processing

Typically words have multiple meanings and multiple forms. As humans, we have a remarkable ability to correctly parse and correctly recognise words strung together in utterances. In the interpretive phase of comprehension it is desirable to quickly narrow the functions and meanings of words. Morphemic analysis helps in this first phase. For example consider the multiple forms of the word drink. If the form "drinking" appears in an utterance, it can be regarded as a participle of a verb, as a noun, or as an adjective. The form "drinkings" can only be regarded as a noun. The following sample output shows how this first small step in the interpretive phase utilizes morphological analysis to aid comprehension by narrowing a word's function based on its morphology.

```
# R NEW:MACLISP
# 22:39.27
= (RESTORE 'CHKPT)
=     NIL
= (VERBOS NIL)
=     NIL
= (OTLL 64)
=     NIL
=
= (CLASS 'DRINK)
=      (N (NS ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))) A (PRES
=      ((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A P1 P2))
=      ((0 0) (/*DRINK3 P1 P2))))
=
= (CLASS 'DRINKS)
=      (N (NP ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))) A (PRES
=     TPS ((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A P1 P2)
=     ) ((0 0) (/*DRINK3 P1 P2))))
=
= (CLASS 'DRINKER)
=      (N (PERS ((0 0) (/*DRINK5))))
```

```
=
= (CLASS 'DRINKERS)
=     (N (NP PERS ((0 0) (/*DRINK5))))
=
= (CLASS 'DRINKING)
=     (N (NS ((0 0) (/*DRINK6))) A (PART ((0 0) (/*DRINK1
=     P1 P2)) ((0 0) (/*DRINK1A P1 P2)) ((0 0) (/*DRINK3
=     P1 P2))) NM (ADJ CLASF ((0 0) (/*DRINK2 P1 P2)) ((0
=      0) (/*DRINK4 P1 P2)) ((0 0) (/*DRINK5 P1 P2)) ((0
=     0) (/*DRINK6 P1 P2))))
=
= (CLASS 'DRINKINGS)
=     (N (NP NS ((0 0) (/*DRINK6))))
=
= (CLASS 'DRINKABLE)
=     (N (NS ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))) NM (ADJ
=     CLASF ((0 0) (/*DRINK2 P1 P2)) ((0 0) (/*DRINK4 P1
=     P2)) ((0 0) (/*DRINK5 P1 P2)) ((0 0) (/*DRINK6 P1 P2)
=     )))
=
= (CLASS 'DRINKABLES)
=     (N (NP NS ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))))
=
= (CLASS 'DRINKWISE)
=     (AM (AT3 ((0 0) (/*DRINK1 KIND)) ((0 0) (/*DRINK1A
=     KIND))))
=
= (CLASS 'DRINKLIKE)
=     (NM (ADJ ((0 0) (/*DRINK2 P1 P2)) ((0 0) (/*DRINK4
=     P1 P2)) ((0 C) (/*DRINK5 P1 P2)) ((0 0) (/*DRINK6 P1
=     P2))))
=
= (CLASS 'DRINKETTE)
=     (N (DIM ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))))
=
= (CLASS 'DRINKETTES)
=     (N (NP DIM ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))))
=
= (CLASS 'DRINKIE)
=     (N (DIM ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))))
=
= (MTS)
```

The following sample listing shows a trace of selected functions that perform parsing and the creation of semantic networks fcr utterances. This example is based on the discussion in Chapter 6 and corresponds to sentence <20> of

Appendix D. The semantic network created for the sentence is shown graphically as part of Figure D10. Note especially the template switch as discussed in Chapter 6.

```
*** ARGUMENTS OF ONTLEDEN
    (((AM ((ADT)) (/*THEN)) (N (NS ((0 0) (/*JUDY1))))
    (A (IREG PAST ((0 0) (/*GIVE1)))) (DET ((NS NP COLL)
    ) (/*THE)) (NM (ADJ CLASF ((0 0) (/*BROWN1)))) (N (
    NS ((0 0) (/*BOOK1)))) (PREP NIL (/*TO1)) (N (NS ((
    0 0) (/*MARY1))))))

    +++ THE CLASSIFIED UTTERANCE IS +++

    ((AM ((ADT)) (/*THEN)) (N (NS ((0 0) (/*JUDY1)))) (
    A (IREG PAST ((0 0) (/*GIVE1)))) (DET ((NS NP COLL)
    ) (/*THE)) (NM (ADJ CLASF ((0 0) (/*BROWN1)))) (N (
    NS ((0 0) (/*BOOK1)))) (PREP NIL (/*TO1)) (N (NS ((
    0 0) (/*MARY1)))))


*** ARGUMENTS OF SENSE
    (((AM ((ADT)) (/*THEN)) (N (NS ((0 0) (/*JUDY1)))))
     (A (IREG PAST ((0 0) (/*GIVE1)))) ((DET ((NS NP COLL)
    ) (/*THE)) (NM (ADJ CLASF ((0 0) (/*BROWN1)))) (N (
    NS ((0 0) (/*BOOK1)))) (PREP NIL (/*TO1)) (N (NS ((
    0 0) (/*MARY1))))) ((0 0) (/*GIVE1)))

*** ARGUMENTS OF SATISFY
    (X /*GIVE1 ((AM ((ADT)) (/*THEN)) (N (NS ((0 0) (/*JUDY1)
    )))))

*** ARGUMENTS OF MATCH
    ((P1) /*GIVE1 X ((AM ((ADT)) (/*THEN)) (N (NS ((0 0
    ) (/*JUDY1))))))

*** ARGUMENTS OF MATCH1
    ((ANIM X) ((AM ((ADT)) (/*THEN)) (N (NS ((0 0) (/*JUDY1)
    )))))

*** VALUE OF MATCH1
    /*JUDY1

*** ARGUMENTS OF MATCH2
    ((P1) /*JUDY1 /*GIVE1)

*** VALUE OF MATCH2
    T

*** VALUE OF MATCH
    (T NIL)
```

```
*** VALUE OF SATISFY
    (T NIL)

*** ARGUMENTS OF SATISFY
    (Y /*GIVE1 ((DET ((NS NP COLL)) (/*THE)) (NM (ADJ CLASF
    ((0 0) (/*BROWN1)))) (N (NS ((0 0) (/*BOOK1)))) (PREP
    NIL (/*TO1)) (N (NS ((0 0) (/*MARY1))))))

*** ARGUMENTS OF MATCH
    ((P2) /*GIVE1 Y ((DET ((NS NP COLL)) (/*THE)) (NM (
    ADJ CLASF ((0 0) (/*BROWN1)))) (N (NS ((0 0) (/*BOOK1)
    ))) (PREP NIL (/*TO1)) (N (NS ((0 0) (/*MARY1))))))


*** ARGUMENTS OF MATCH1
    ((ANIM Y) ((DET ((NS NP COLL)) (/*THE)) (NM (ADJ CLASF
    ((0 0) (/*BROWN1)))) (N (NS ((0 0) (/*BOOK1)))) (PREP
    NIL (/*TO1)) (N (NS ((0 0) (/*MARY1))))))

*** VALUE OF MATCH1
    /*MARY1

*** ARGUMENTS OF MATCH2
    ((P2) /*MARY1 /*GIVE1)

*** VALUE OF MATCH2
    T

*** VALUE OF MATCH
    (T NIL)

*** VALUE OF SATISFY
    (T NIL)

*** ARGUMENTS OF SATISFY
    (Z /*GIVE1 NIL)

*** VALUE OF SATISFY
    NIL

*** ARGUMENTS OF SATISFY
    (X /*GIVE1 ((AM ((ADT)) (/*THEN)) (N (NS ((0 0) (/*JUDY1)
    )))))

*** ARGUMENTS OF MATCH
    ((P1) /*GIVE1 X ((AM ((ADT)) (/*THEN)) (N (NS ((0 0
    ) (/*JUDY1))))))

*** ARGUMENTS OF MATCH1
    ((ANIM X) ((AM ((ADT)) (/*THEN)) (N (NS ((0 0) (/*JUDY1)
    )))))
```

```
*** VALUE OF MATCH1
    /*JUDY1

*** ARGUMENTS OF MATCH2
    ((P1) /*JUDY1 /*GIVE1)

*** VALUE OF MATCH2
    T

*** VALUE OF MATCH
    (T NIL)

*** VALUE OF SATISFY
    (T NIL)

*** ARGUMENTS OF SATISFY
    (Z /*GIVE1 ((DET ((NS NP COLL)) (/*THE)) (NM (ADJ CLASF
    ((0 0) (/*BROWN1)))) (N (NS ((0 0) (/*BOOK1)))) (PREP
    NIL (/*TO1)) (N (NS ((0 0) (/*MARY1))))))

*** ARGUMENTS OF MATCH
    ((P3) /*GIVE1 Z ((DET ((NS NP COLL)) (/*THE)) (NM (
    ADJ CLASF ((0 0) (/*BROWN1)))) (N (NS ((0 0) (/*BOOK1)
    ))) (PREP NIL (/*TO1)) (N (NS ((0 0) (/*MARY1))))))


*** ARGUMENTS OF MATCH1
    ((INANIM Z) ((DET ((NS NP COLL)) (/*THE)) (NM (ADJ
    CLASF ((0 0) (/*BROWN1)))) (N (NS ((0 0) (/*BOOK1))
    )) (PREP NIL (/*TO1)) (N (NS ((0 0) (/*MARY1))))))

*** VALUE OF MATCH1
    /*BOOK1

*** ARGUMENTS OF MATCH2
    ((P3) /*BOOK1 /*GIVE1)

*** VALUE OF MATCH2
    T

*** VALUE OF MATCH
    (T ((PREP NIL (/*TO1)) (N (NS ((0 0) (/*MARY1))))))


*** VALUE OF SATISFY
    (T ((PREP NIL (/*TO1)) (N (NS ((0 0) (/*MARY1))))))


*** ARGUMENTS OF SATIS
    (/*TO1 ((PREP NIL (/*TO1)) (N (NS ((0 0) (/*MARY1))
    ))))

*** VALUE OF SATIS
```

```
     ((N (NS ((0 0) (/*MARY1)))))

*** ARGUMENTS OF SATISFY
    (Y /*GIVE1 ((N (NS ((0 0) (/*MARY1))))))

*** ARGUMENTS OF MATCH
    ((P2) /*GIVE1 Y ((N (NS ((0 0) (/*MARY1))))))

*** ARGUMENTS OF MATCH1
    ((ANIM Y) ((N (NS ((0 0) (/*MARY1))))))

*** VALUE OF MATCH1
    /*MARY1

*** ARGUMENTS OF MATCH2
    ((P2) /*MARY1 /*GIVE1)

*** VALUE OF MATCH2
    T

*** VALUE OF MATCH
    (T NIL)

*** VALUE OF SATISFY
    (T NIL)

*** VALUE OF SENSE
    T

*** VALUE OF ONTLEDEN
    (/*GIVE1 (((/*GIVE1 /*MARY1 Y) (/*GIVE1 /*BOOK1 Z)
    (/*GIVE1 /*JUDY1 X)) ((/*GIVE1 /*BOOK1 Z) (/*GIVE1
    /*JUDY1 Y) (/*GIVE1 /*ANNE1 X))))

*** ARGUMENTS OF NET
    ((/*GIVE1 (((/*GIVE1 /*MARY1 Y) (/*GIVE1 /*BOOK1 Z)
     (/*GIVE1 /*JUDY1 X)) ((/*GIVE1 /*BOOK1 Z) (/*GIVE1
    /*JUDY1 Y) (/*GIVE1 /*ANNE1 X)))))
    +++ ASSOCIATED <ACTICN-ARGUMENT-VARIABLE> TRIPLES +++

     ((*GIVE1 *MARY1 Y) (*GIVE1 *BOOK1 Z) (*GIVE1 *JUDY1
    X))

    +++MODIFIERS+++

     ((NM (ADJ CLASF ((0 0) (*BROWN1)))) Z)

    PREDICATIVES->
     NIL
    ADJECTIVES->
     ((NM (ADJ CLASF ((0 0) (*BROWN1)))) Z)
    EXPLICIT-ADJECTIVES->
    IMPLICIT-ADJECTIVES->
```

```
    OPERATORS->
     NIL

*** ARGUMENTS OF CREATENET
    (/*GIVE1 ((/*GIVE1 /*JUDY1 X) (/*GIVE1 /*BOOK1 Z) (
    /*GIVE1 /*MARY1 Y) (/*GIVE1 /*BROWN1 Z)))

         +++ THE SEMANTIC NET +++
      *ATOM*      *VALUE*     *PROPERTY*

    PROP0150     *JUDY1      X
    PROP0151     *BOCK1      PRED
    PROP0151     INST0152      ARG
    PROP0153     INST0152      ARG
    PROP0153     *UNS0154      PRED
    PROP0150     INST0152      Z
    PROP0150     *MARY1      Y
    PROP0155     *BRCWN1      PRED
    PROP0155     INST0152      ARG
    PROP0150     *GIVE1      PRED

*** VALUE OF CREATENET
    PROP0150

*** VALUE OF NET
    NIL
    NIL
```

Appendix I   Examples of Memory Nets

The following is a sample run of the experimental program showing the semantic networks built for the sentences below. The sentences are numbered <1> through <20> and following the output listing are figures corresponding to each (UNDERSTAND) invocation. Note well that I am using descriptions in the attributive sense of Russell for illustrative pruposes. It remains a major problem to use and formulate descriptions referentially.

```
$R NEW:MACLISP
23:55.40
=>                      (RESTORE 'CHKPT)
    NIL
=>                      (VERBOS NIL)
    NIL
=>                      (OTLL 64)
    NIL
=>                      (INLL 255)
    NIL
=>                      (SETQ FULLNET 'ON)
    ON
=>                      (STARTUP)

                  HELLO AND WELCOME TO THE
     *** STATE-BASED CONCEPTUAL REPRESENTATION SYSTEM ***

      TO INITIATE DIALOG, TYPE:

       (UNDERSTAND)

       WHEN THE SYSTEM IS READY, TYPE IN
       ANY DIALOG FOLLOWED BY TWO CARRAIGE
       RETURNS. A STRUCTURE WILL BE BUILT
       AND THEN YOU CAN CONTINUE BY TYPING
       (UNDERSTAND), ETC.


    NIL
=>                      (UNDERSTAND)
    READY
```

```
=>                         JOHN IS A MAN.                         <1>
=>                         JUDY IS A BIG WOMAN.                   <2>
=>                         MARY IS SHORT.                         <3>
=>
=>


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((N (NS ((0 0) (/*JOHN1)))) (A (IREG PRES ((0 0) (/*EQ1)
    ))) (DET ((NS INDEF)) (/*A)) (N (NS ((0 0) (/*MAN1)
    ))))



    +++ THE CLASSIFIED UTTERANCE IS +++

    ((N (NS ((0 0) (/*JOHN1)))) (DET ((NS INDEF)) (/*A)
    ) (N (NS ((0 0) (/*MAN1))))))


    +++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

     ((*EQ1 *MAN1 Y) (*EQ1 *JOHN1 X))

    +++MODIFIERS+++


    PREDICATIVES->
     NIL

         +++ THE SEMANTIC NET +++
      *ATOM*       *VALUE*       *PROPERTY*

    PROP0001       *JOHN1       X
    *JOHN1       PROP0001       PROPOSITIONS
    PROP0001       *MAN1       PRED
    *MAN1       PROP0001       PROPOSITIONS


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((N (NS ((0 0) (/*JUDY1)))) (A (IREG PRES ((0 0) (/*EQ1)
    ))) (DET ((NS INDEF)) (/*A)) (NM (ADJ CLASF ((0 0)
    (/*BIG1)))) (N (NS ((0 0) (/*WOMAN1)))))



    +++ THE CLASSIFIED UTTERANCE IS +++

    ((N (NS ((0 0) (/*JUDY1)))) (DET ((NS INDEF)) (/*A)
    ) (NM (ADJ CLASF ((0 0) (/*BIG1)))) (N (NS ((0 0) (
```

```
/*WOMAN1)))))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*EQ1 *WOMAN1 Y) (*EQ1 *JUDY1 X))

+++MODIFIERS+++

 ((NM (ADJ CLASF ((0 0) (*BIG1)))) Y)

PREDICATIVES->
 NIL
ADJECTIVES->
EXPLICIT-ADJECTIVES->
IMPLICIT-ADJECTIVES->
 ((NM (ADJ CLASF ((0 0) (*BIG1)))) Y)
OPERATORS->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

*WOMAN1      PROP0003      PROPOSITIONS
PROP0003      *WOMAN1      PRED
PROP0003      INST0004      ARG
INST0004      PROP0002
               PROP0003      PROPOSITIONS
PROP0002      INST0004      Y
PROP0002      *JUDY1      X
*JUDY1      PROP0002      PROPOSITIONS
PROP0002      *EQ1      PRED
*EQ1      PROP0002      PROPOSITIONS
PROP0002      *GREATER1      PRED
*GREATER1      PROP0002      PROPOSITIONS
PROP0002      INST0005      Y
INST0005      PROP0002      PROPOSITIONS
INST0005      *WOMAN1      CONCEPT
INST0005      *TYPVAL*      FUNC
INST0005      *SIZE1      MEASATTR
*TYPVAL*      INST0005      PROPOSITIONS
*WOMAN1      INST0005
               PROP0003      PROPOSITIONS
*SIZE1      INST0005      PROPOSITIONS
*JUDY1      PROP0003
               PROP0002      PROPOSITIONS

REMOVED PROPERTY: ARG   FROM  PROP0003

PROP0003      *JUDY1      ARG
INST0006      *JUDY1      X
*JUDY1      INST0006
               PROP0003
```

```
                 PROP0002     PROPOSITIONS
PROP0002     INST0006     X
INST0006     PROP0002     PROPOSITIONS
INST0006     *SIZE1     FUNC
*SIZE1     INST0006
                 INST0005     PROPOSITIONS
```

+++ THE CLASSIFIED UTTERANCE IS +++

```
((N (NS ((0 0) (/*MARY1)))) (A (IREG PRES ((0 0) (/*EQ1)
))) (NM (ADJ CLASF ((0 0) (/*SHORT1))))))
```

+++ THE CLASSIFIED UTTERANCE IS +++

```
((N (NS ((0 0) (/*MARY1)))) (NM (ADJ CLASF ((0 0) (
/*SHORT1))))))
```

+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

```
 ((*EQ1 *SHORT1 Y) (*EQ1 *MARY1 X))
```

+++MODIFIERS+++

```
PREDICATIVES->
 ((NM (ADJ CLASF ((0 0) (*SHORT1)))) Y)
```

+++ THE SEMANTIC NET +++

```
  *ATOM*      *VALUE*      *PROPERTY*

*SOMETHING1      PROP0008      PROPOSITIONS
PROP0008      *SOMETHING1      PRED
PROP0008      INST0009      ARG
INST0009      PROP0007
                 PROP0008      PROPOSITIONS
PROP0007      INST0009      Y
PROP0007      *MARY1      X
*MARY1      PROP0007      PROPOSITIONS
PROP0007      *EQ1      PRED
*EQ1      PROP0007
                 PROP0002      PROPOSITIONS
PROP0007      *LESS1      PRED
*LESS1      PROP0007      PROPOSITIONS
PROP0007      INST0010      Y
INST0010      PROP0007      PROPOSITIONS
INST0010      *SOMETHING1      CONCEPT
INST0010      *TYPVAL*      FUNC
INST0010      *HEIGHT1      MEASATTR
*TYPVAL*      INST0010
```

```
                    INST0005     PROPOSITIONS
    *SOMETHING1       INST0010
                    PROP0008     PROPOSITIONS
    *HEIGHT1     INST0010     PROPOSITIONS
    *MARY1     PROP0008
                    PROP0007     PROPOSITIONS

    REMOVED PROPERTY: ARG   FROM  PROP0008

    PROP0008     *MARY1     ARG
    INST0011     *MARY1     X
    *MARY1     INST0011
                    PROP0008
                    PROP0007     PROPOSITIONS
    PROP0007     INST0011     X
    INST0011     PROP0007     PROPOSITIONS
    INST0011     *HEIGHT1     FUNC
    *HEIGHT1     INST0011
                    INST0010     PROPOSITIONS
    NIL
=>
=>                     (SETQ FULLNET NIL)
    NIL
=>
=>                     (UNDERSTAND)
    READY

=>                     JOHN IS BIGGER THAN BILL.              <4>
=>                     BUT, BILL IS HEAVIER THAN JOHN.        <5>
=>
=>


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((N  (NS  ((0  0)  (/*JOHN1))))  (A  (IREG PRES  ((0  0)  (/*EQ1)
    )))  (NM (ADJ COM CLASF ((0  0)  (/*BIG1))))  (BIND NIL
    (/*THAN))  (N  (NS  ((0  0)  (/*BILL1)))))


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((N  (NS  ((0  0)  (/*JOHN1))))  (NM  (ADJ COM CLASF  ((0
    0)  (/*BIG1))))  (BIND NIL (/*THAN))  (N  (NS  ((0  0)  (/*BILL1)
    ))))


    +++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

     ((*EQ1 *BILL1 Y)  (*EQ1 *JOHN1 X))

    +++MODIFIERS+++
```

```
 ((NM (ADJ COM CLASF ((0 0) (*BIG1)))) Y)

PREDICATIVES->
 NIL
ADJECTIVES->
EXPLICIT-ADJECTIVES->
 ((NM (ADJ COM CLASF ((0 0) (*BIG1)))) Y)
IMPLICIT-ADJECTIVES->
OPERATORS->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*     *VALUE*     *PROPERTY*

PROP0012     *BILL1      Y
PROP0012     *JOHN1      X
PROP0012     *EQ1      PRED
PROP0012     *GREATER1      PRED
INST0013     *JOHN1      X
INST0014     *BILL1      X
PROP0012     INST0013      X
PROP0012     INST0014      Y
INST0013     *SIZE1      FUNC
INST0014     *SIZE1      FUNC
*SIZE1     INST0013      VALUE
*SIZE1     INST0014
              INST0013      VALUE


+++ THE CLASSIFIED UTTERANCE IS +++

((BIND NIL (/*BUT1)) (N (NS ((0 0) (/*BILL1)))) (A
(IREG PRES ((0 0) (/*EQ1)))) (NM (ADJ COM CLASF ((0
 0) (/*HEAVY1)))) (BIND NIL (/*THAN)) (N (NS ((0 0)
 (/*JOHN1)))))



+++ THE CLASSIFIED UTTERANCE IS +++

((BIND NIL (/*BUT1)) (N (NS ((0 0) (/*BILL1)))) (NM
(ADJ COM CLASF ((0 0) (/*HEAVY1)))) (BIND NIL (/*THAN)
) (N (NS ((0 0) (/*JOHN1)))))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*EQ1 *JOHN1 Y) (*EQ1 *BILL1 X))

+++MODIFIERS+++

 ((NM (ADJ COM CLASF ((0 0) (*HEAVY1)))) Y)
```

```
PREDICATIVES->
 NIL
ADJECTIVES->
EXPLICIT-ADJECTIVES->
 ((NM (ADJ COM CLASF ((0 0) (*HEAVY1)))) Y)
IMPLICIT-ADJECTIVES->
OPERATORS->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

PROP0015      *JOHN1       Y
PROP0015      *BILL1       X
PROP0015      *EQ1     PRED
PROP0015      *GREATER1      PRED
INST0016      *BILL1       X
INST0017      *JOHN1       X
PROP0015      INST0016       X
PROP0015      INST0017       Y
INST0016      *WEIGHT1       FUNC
INST0017      *WEIGHT1       FUNC
*WEIGHT1      INST0016       VALUE
*WEIGHT1      INST0017
               INST0016       VALUE
NIL
=>
=>                     (UNDERSTAND)
READY

=>                    BIG TALL MARK IS DRINKING THE UNSAVORY    <6>
=>                    PINK JUICE.  THAT MAN DRINKS THE          <7>
=>                    WHISKEY. THAT MAN DRINKS THE DELICIOUS    <8>
=>                    COFFEE. MARY ALSO IS DRINKING COFFEE.     <9>
=>
=>


+++ THE CLASSIFIED UTTERANCE IS +++

((NM (ADJ CLASF ((0 0) (/*BIG1)))) (NM (ADJ ((0 0)
(/*TALL1)))) (N (NS ((0 0) (/*MARK1)))) (A (IREG PRES
((0 0) (/*EQ1)))) (N (NS ((0 0) (/*DRINK6))) A (PART
((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A P1 P2))
((0 0) (/*DRINK3 P1 P2))) NM (ADJ CLASF ((0 0) (/*DRINK2
P1 P2)) ((0 0) (/*DRINK4 P1 P2)) ((0 0) (/*DRINK5 P1
P2)) ((0 0) (/*DRINK6 P1 P2)))) (DET ((NS NP COLL))
 (/*THE)) (NM (ADJ CLASF ((0 0) (/*UNSAVORY)))) (NM
(ADJ CLASF ((0 0) (/*PINK1)))) (N (NS ((0 0) (/*JUICE1)
))))
```

```
+++ THE CLASSIFIED UTTERANCE IS +++

((NM (ADJ CLASF ((0 0) (/*BIG1)))) (NM (ADJ ((0 0)
(/*TALL1)))) (N (NS ((0 0) (/*MARK1)))) (N (NS ((0
0) (/*DRINK6))) A (PART ((0 0) (/*DRINK1 P1 P2)) ((
0 0) (/*DRINK1A P1 P2)) ((0 0) (/*DRINK3 P1 P2))) NM
(ADJ CLASF ((0 0) (/*DRINK2 P1 P2)) ((0 0) (/*DRINK4
P1 P2)) ((0 0) (/*DRINK5 P1 P2)) ((0 0) (/*DRINK6 P1
P2)))) (DET ((NS NP COLL)) (/*THE)) (NM (ADJ CLASF
((0 0) (/*UNSAVORY)))) (NM (ADJ CLASF ((0 0) (/*PINK1)
))) (N (NS ((0 0) (/*JUICE1)))))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*DRINK1 *JUICE1 Y) (*DRINK1 *MARK1 X))

+++MODIFIERS+++

 ((NM (ADJ CLASF ((0 0) (*PINK1)))) Y)
 ((NM (ADJ CLASF ((0 0) (*UNSAVORY)))) Y)
 ((NM (ADJ ((0 0) (*TALL1)))) X)
 ((NM (ADJ CLASF ((0 0) (*BIG1)))) X)

PREDICATIVES->
 NIL
ADJECTIVES->
 ((NM (ADJ CLASF ((0 0) (*UNSAVORY)))) Y)
 ((NM (ADJ CLASF ((0 0) (*PINK1)))) Y)
EXPLICIT-ADJECTIVES->
IMPLICIT-ADJECTIVES->
 ((NM (ADJ CLASF ((0 0) (*BIG1)))) X)
 ((NM (ADJ ((0 0) (*TALL1)))) X)
OPERATORS->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*     *VALUE*     *PROPERTY*

PROP0018     *MARK1       X
PROP0019     *JUICE1      PRED
PROP0019     INST0020      ARG
PROP0021     INST0020      ARG
PROP0021     *UNS0022      PRED
PROP0018     INST0020      Y
PROP0023     *UNSAVORY      PRED
PROP0023     INST0020      ARG
PROP0024     *PINK1       PRED
PROP0024     INST0020      ARG
PROP0018     *DRINK1      PRED
PROP0025     *GREATER1      PRED
INST0026     *SIZE1       MEASATTR
```

```
INST0026      *TYPVAL*      FUNC
INST0026      *SOMETHING1      CONCEPT
PROP0027      *SOMETHING1      PRED
PROP0027      *MARK1      ARG
PROP0025      INST0026      Y
PROP0025      INST0028      X
INST0028      *SIZE1      FUNC
INST0028      *MARK1      X
PROP0029      *GREATER1      PRED
INST0030      *HEIGHT1      MEASATTR
INST0030      *TYPVAL*      FUNC
INST0030      *SOMETHING1      CONCEPT
PROP0029      INST0030      Y
PROP0029      INST0031      X
INST0031      *HEIGHT1      FUNC
INST0031      *MARK1      X


+++ THE CLASSIFIED UTTERANCE IS +++

((PRO ((DEF DEM REL)) (/*THAT)) (N (NS ((0 0) (/*MAN1)
))) (N (NP ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))) A
(PRES TPS ((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A
P1 P2)) ((0 0) (/*DRINK3 P1 P2)))) (DET ((NS NP COLL)
) (/*THE)) (N (NS ((0 0) (/*WHISKEY1)))))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*DRINK1A *WHISKEY1 Y) (*DRINK1A *MAN1 X))

+++MODIFIERS+++


PREDICATIVES->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

PROP0033      *WHISKEY1      PRED
PROP0033      INST0034      ARG
PROP0035      INST0034      ARG
PROP0035      *UNS0036      PRED
PROP0032      INST0034      Y
PROP0037      *MAN1      PRED
PROP0037      INST0038      ARG
PROP0039      INST0038      ARG
PROP0039      *UNS0040      PRED
PROP0032      INST0038      X
PROP0032      *DRINK1A      PRED
```

```
+++ THE CLASSIFIED UTTERANCE IS +++

((PRO ((DEF DEM REL)) (/*THAT)) (N (NS ((0 0) (/*MAN1)
))) (N (NP ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))) A
(PRES TPS ((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A
P1 P2)) ((0 0) (/*DRINK3 P1 P2)))) (DET ((NS NP COLL)
) (/*THE)) (NM (ADJ ((0 0) (/*DELICIOUS1)))) (N (NS
((0 0) (/*COFFEE1)))))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*DRINK1 *COFFEE1 Y) (*DRINK1 *MAN1 X))

+++MODIFIERS+++

 ((NM (ADJ ((0 0) (*DELICIOUS1)))) Y)

PREDICATIVES->
 NIL
ADJECTIVES->
 ((NM (ADJ ((0 0) (*DELICIOUS1)))) Y)
EXPLICIT-ADJECTIVES->
IMPLICIT-ADJECTIVES->
OPERATORS->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

PROP0042      *MAN1      PRED
PROP0042      INST0043      ARG
PROP0044      INST0043      ARG
PROP0044      *UNS0045      PRED
PROP0041      INST0043      X
PROP0046      *COFFEE1      PRED
PROP0046      INST0047      ARG
PROP0048      INST0047      ARG
PROP0048      *UNS0049      PRED
PROP0041      INST0047      Y
PROP0050      *DELICIOUS1      PRED
PROP0050      INST0047      ARG
PROP0041      *DRINK1      PRED


+++ THE CLASSIFIED UTTERANCE IS +++

((N (NS ((0 0) (/*MARY1)))) NIL (A (IREG PRES ((0 0
) (/*EQ1)))) (N (NS ((0 0) (/*DRINK6))) A (PART ((0
 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A P1 P2)) ((0
 0) (/*DRINK3 P1 P2))) NM (ADJ CLASF ((0 0) (/*DRINK2
P1 P2)) ((0 0) (/*DRINK4 P1 P2)) ((0 0) (/*DRINK5 P1
P2)) ((0 0) (/*DRINK6 P1 P2)))) (N (NS ((0 0) (/*COFFEE1)
```

```
))))


+++ THE CLASSIFIED UTTERANCE IS +++

((N (NS ((0 0) (/*MARY1)))) NIL (N (NS ((0 0) (/*DRINK6)
)) A (PART ((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A
P1 P2)) ((0 0) (/*DRINK3 P1 P2))) NM (ADJ CLASF ((0
 0) (/*DRINK2 P1 P2)) ((0 0) (/*DRINK4 P1 P2)) ((0
0) (/*DRINK5 P1 P2)) ((0 0) (/*DRINK6 P1 P2)))) (N
(NS ((0 0) (/*COFFEE1)))))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*DRINK1 *COFFEE1 Y) (*DRINK1 *MARY1 X))

+++MODIFIERS+++


PREDICATIVES->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

PROP0052      *COFFEE1      PRED
PROP0052      INST0053      ARG
PROP0054      INST0053      ARG
PROP0054      *UNS0055      PRED
PROP0051      INST0053      Y
PROP0051      *MARY1      X
PROP0051      *DRINK1      PRED
NIL
=>
=>                 (UNDERSTAND)
READY

=>           THE ACID IS EATING MY CAR AWAY TO NOTHING. <10>
=>           MY YELLOW CAR IS DRINKING HEAVY
=>           GASOLINE.                                  <11>
=>
=>


+++ THE CLASSIFIED UTTERANCE IS +++

((DET ((NS NP COLL)) (/*THE)) (N (NS ((0 0) (/*ACID1)
))) (A (IREG PRES ((0 0) (/*EQ1)))) (A (PART ((0 0)
 (/*EAT1 P1 P2)) ((0 0) (/*EAT2 P1 P2))) NM (ADJ CLASF
((0 0) (/*EAT3)))) (PRO ((DEF PERS NS POSS)) (/*I))
 (N (NS ((0 0) (/*CAR1)))) (PRT NIL (/*AWAY)) (PREP
```

```
NIL (/*TO1)) (PRO ((INDEF NS)) (/*NOTHING1)))


+++ THE CLASSIFIED UTTERANCE IS +++

((DET ((NS NP COLL)) (/*THE)) (N (NS ((0 0) (/*ACID1)
))) (A (PART ((0 0) (/*EAT1 P1 P2)) ((0 0) (/*EAT2
P1 P2))) NM (ADJ CLASF ((0 0) (/*EAT3)))) (PRO ((DEF
PERS NS POSS)) (/*I)) (N (NS ((0 0) (/*CAR1)))) (PRT
NIL (/*AWAY)) (PREP NIL (/*TO1)) (PRO ((INDEF NS))
(/*NOTHING1)))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*EAT2 *CAR1 Y) (*EAT2 *ACID1 X))

+++MODIFIERS+++


PREDICATIVES->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

PROP0057      *CAR1      PRED
PROP0057      INST0058      ARG
PROP0059      INST0058      ARG
PROP0059      *UNSC060      PRED
PROP0056      INST0058      Y
PROP0061      *ACID1      PRED
PROP0061      INST0062      ARG
PROP0063      INST0062      ARG
PROP0063      *UNSC064      PRED
PROP0056      INST0062      X
PROP0056      *EAT2      PRED


+++ THE CLASSIFIED UTTERANCE IS +++

((PRO ((DEF PERS NS POSS)) (/*I)) (NM (ADJ CLASF ((
0 0) (/*YELLOW1)))) (N (NS ((0 0) (/*CAR1)))) (A (IREG
PRES ((0 0) (/*EQ1)))) (N (NS ((0 0) (/*DRINK6))) A
(PART ((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A P1
P2)) ((0 0) (/*DRINK3 P1 P2))) NM (ADJ CLASF ((0 0)
 (/*DRINK2 P1 P2)) ((0 0) (/*DRINK4 P1 P2)) ((0 0)
(/*DRINK5 P1 P2)) ((0 0) (/*DRINK6 P1 P2)))) (NM (ADJ
CLASF ((0 0) (/*HEAVY1)))) (N (NS ((0 0) (/*GAS1)))
))
```

```
+++ THE CLASSIFIED UTTERANCE IS +++

((PRO ((DEF PERS NS POSS)) (/*I)) (NM (ADJ CLASF ((
0 0) (/*YELLOW1)))) (N (NS ((0 0) (/*CAR1)))) (N (NS
((0 0) (/*DRINK6))) A (PART ((0 0) (/*DRINK1 P1 P2)
) ((0 0) (/*DRINK1A P1 P2)) ((0 0) (/*DRINK3 P1 P2)
)) NM (ADJ CLASF ((0 0) (/*DRINK2 P1 P2)) ((0 0) (/*DRINK4
P1 P2)) ((0 0) (/*DRINK5 P1 P2)) ((0 0) (/*DRINK6 P1
P2)))) (NM (ADJ CLASF ((0 0) (/*HEAVY1)))) (N (NS (
(0 0) (/*GAS1)))))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*DRINK3 *GAS1 Y) (*DRINK3 *CAR1 X))

+++MODIFIERS+++

 ((NM (ADJ CLASF ((0 0) (*HEAVY1)))) Y)
 ((NM (ADJ CLASF ((0 0) (*YELLOW1)))) X)

PREDICATIVES->
 NIL
ADJECTIVES->
 ((NM (ADJ CLASF ((0 0) (*YELLOW1)))) X)
EXPLICIT-ADJECTIVES->
IMPLICIT-ADJECTIVES->
 ((NM (ADJ CLASF ((0 0) (*HEAVY1)))) Y)
OPERATORS->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*     *PROPERTY*

PROP0066      *CAR1      PRED
PROP0066      INST0067      ARG
PROP0068      INST0067      ARG
PROP0068      *UNS0069      PRED
PROP0065      INST0067      X
PROP0070      *GAS1      PRED
PROP0070      INST0071      ARG
PROP0072      INST0071      ARG
PROP0072      *UNS0073      PRED
PROP0065      INST0071      Y
PROP0074      *YELLOW1      PRED
PROP0074      INST0067      ARG
PROP0065      *DRINK3     PRED
PROP0075      *GREATER1      PRED
INST0076      *WEIGHT1      MEASATTR
INST0076      *TYPVAL*      FUNC
INST0076      *GAS1     CONCEPT
PROP0075      INST0076      Y
```

```
    PROP0075     INST0077     X
    INST0077     *WEIGHT1     FUNC
    INST0077     INST0071     X
    NIL
=>
=>                    (UNDERSTAND)
    READY

=>                    SHORT, SMALL ANNE IS EATING THE DELICIOUS,
=>                    LIGHT, ROUND, YELLOW CAKE.          <12>
=>
=>


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((NM (ADJ CLASF ((0 0) (/*SHORT1)))) (NM (ADJ CLASF
    ((0 0) (/*SMALL1)))) (N (NS ((0 0) (/*ANNE1)))) (A
    (IREG PRES ((0 0) (/*EQ1)))) (A (PART ((0 0) (/*EAT1
    P1 P2)) ((0 0) (/*EAT2 P1 P2))) NM (ADJ CLASF ((0 0
    ) (/*EAT3)))) (DET ((NS NP COLL)) (/*THE)) (NM (ADJ
    ((0 0) (/*DELICIOUS1)))) (NM (ADJ CLASF ((0 0) (/*LIGHT1)
    ))) (NM (ADJ ((0 0) (/*ROUND1)))) (NM (ADJ CLASF ((
    0 0) (/*YELLOW1)))) (N (NS ((0 0) (/*CAKE1)))))


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((NM (ADJ CLASF ((0 0) (/*SHORT1)))) (NM (ADJ CLASF
    ((0 0) (/*SMALL1)))) (N (NS ((0 0) (/*ANNE1)))) (A
    (PART ((0 0) (/*EAT1 P1 P2)) ((0 0) (/*EAT2 P1 P2))
    ) NM (ADJ CLASF ((0 0) (/*EAT3)))) (DET ((NS NP COLL)
    ) (/*THE)) (NM (ADJ ((0 0) (/*DELICIOUS1)))) (NM (ADJ
    CLASF ((0 0) (/*LIGHT1)))) (NM (ADJ ((0 0) (/*ROUND1)
    ))) (NM (ADJ CLASF ((0 0) (/*YELLOW1)))) (N (NS ((0
     0) (/*CAKE1)))))


    +++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

     ((*EAT1 *CAKE1 Y) (*EAT1 *ANNE1 X))

    +++MODIFIERS+++

     ((NM (ADJ CLASF ((0 0) (*YELLOW1)))) Y)
     ((NM (ADJ ((0 0) (*ROUND1)))) Y)
     ((NM (ADJ CLASF ((0 0) (*LIGHT1)))) Y)
     ((NM (ADJ ((0 0) (*DELICIOUS1)))) Y)
     ((NM (ADJ CLASF ((0 0) (*SMALL1)))) X)
     ((NM (ADJ CLASF ((0 0) (*SHORT1)))) X)

    PREDICATIVES->
```

```
 NIL
ADJECTIVES->
 ((NM (ADJ ((0 0) (*DELICIOUS1)))) Y)
 ((NM (ADJ ((0 0) (*ROUND1)))) Y)
 ((NM (ADJ CLASF ((0 0) (*YELLOW1)))) Y)
EXPLICIT-ADJECTIVES->
IMPLICIT-ADJECTIVES->
 ((NM (ADJ CLASF ((0 0) (*SHORT1)))) X)
 ((NM (ADJ CLASF ((0 0) (*SMALL1)))) X)
 ((NM (ADJ CLASF ((0 0) (*LIGHT1)))) Y)
OPERATORS->
 NIL

    +++ THE SEMANTIC NET +++
  *ATOM*       *VALUE*      *PROPERTY*

PROP0078     *ANNE1      X
PROP0079     *CAKE1      PRED
PROP0079     INST0080      ARG
PROP0081     INST0080      ARG
PROP0081     *UNS0082      PRED
PROP0078     INST0080      Y
PROP0083     *DELICIOUS1      PRED
PROP0083     INST0080      ARG
PROP0084     *ROUND1      PRED
PROP0084     INST0080      ARG
PROP0085     *YELLOW1      PRED
PROP0085     INST0080      ARG
PROP0078     *EAT1      PRED
PROP0086     *LESS1      PRED
INST0087     *HEIGHT1      MEASATTR
INST0087     *TYPVAL*      FUNC
INST0087     *SOMETHING1      CONCEPT
PROP0088     *SOMETHING1      PRED
PROP0088     *ANNE1      ARG
PROP0086     INST0087      Y
PROP0086     INST0089      X
INST0089     *HEIGHT1      FUNC
INST0089     *ANNE1      X
PROP0090     *LESS1      PRED
INST0091     *SIZE1      MEASATTR
INST0091     *TYPVAL*      FUNC
INST0091     *SOMETHING1      CONCEPT
PROP0090     INST0091      Y
PROP0090     INST0092      X
INST0092     *SIZE1      FUNC
INST0092     *ANNE1      X
PROP0093     *LESS1      PRED
INST0094     *WEIGHT1      MEASATTR
INST0094     *TYPVAL*      FUNC
INST0094     *CAKE1      CONCEPT
PROP0093     INST0094      Y
PROP0093     INST0095      X
```

```
    INST0095     *WEIGHT1     FUNC
    INST0095     INST0080     X
    NIL
=>
=>                     (UNDERSTAND)
    READY

=>                   THE DRINKER IS ONE WHO DRINKS DRINKS.   <13>
=>
=>


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((DET ((NS NP COLL)) (/*THE)) (N (PERS ((0 0) (/*DRINK5)
    ))) (A (IREG PRES ((0 0) (/*EQ1)))) (PRO ((INDEF))
    (/*ONE1)) (PRO ((DEF REL NS NP PERS)) (/*WHO)) (N (
    NP ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4))) A (PRES TPS
    ((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A P1 P2))
    ((0 0) (/*DRINK3 P1 P2)))) (N (NP ((0 0) (/*DRINK2)
    ) ((0 0) (/*DRINK4))) A (PRES TPS ((0 0) (/*DRINK1
    P1 P2)) ((0 0) (/*DRINK1A P1 P2)) ((0 0) (/*DRINK3
    P1 P2)))))


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((DET ((NS NP COLL)) (/*THE)) (N (PERS ((0 0) (/*DRINK5)
    ))) (PRO ((INDEF)) (/*ONE1)) (PRO ((DEF REL NS NP PERS)
    ) (/*WHO)) (N (NP ((0 0) (/*DRINK2)) ((0 0) (/*DRINK4)
    )) A (PRES TPS ((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A
    P1 P2)) ((0 0) (/*DRINK3 P1 P2)))) (N (NP ((0 0) (/*DRINK2)
    ) ((0 0) (/*DRINK4))) A (PRES TPS ((0 0) (/*DRINK1
    P1 P2)) ((0 0) (/*DRINK1A P1 P2)) ((0 0) (/*DRINK3
    P1 P2)))))


    +++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

     ((*DRINK1A *DRINK2 Y) (*DRINK1A *DRINK5 X))

    +++MODIFIERS+++


    PREDICATIVES->
     NIL

        +++ THE SEMANTIC NET +++
      *ATOM*      *VALUE*      *PROPERTY*

    PROP0097     *DRINK2      PRED
    PROP0097     INST0098      ARG
```

```
    PROP0099     INST0098     ARG
    PROP0099     *UNS0100     PRED
    PROP0096     INST0098     Y
    PROP0101     *DRINK5      PRED
    PROP0101     INST0102     ARG
    PROP0103     INST0102     ARG
    PROP0103     *UNS0104     PRED
    PROP0096     INST0102     X
    PROP0096     *DRINK1A     PRED
    NIL
=>
=>                 (UNDERSTAND)
    READY

 =>                 MARY IS PERFECT. JOHN IS AN       <14>
 =>                 IDEAL MAN. BIG MIKE IS A          <15>
 =>                 PERFECT FAT MAN.                  <16>
 =>
 =>


     +++ THE CLASSIFIED UTTERANCE IS +++

     ((N (NS ((0 0) (/*MARY1)))) (A (IREG PRES ((0 0) (/*EQ1)
     ))) (NM (ADJ ((0 0) (/*PERFECT1))))))



     +++ THE CLASSIFIED UTTERANCE IS +++

     ((N (NS ((0 0) (/*MARY1)))) (NM (ADJ ((0 0) (/*PERFECT1)
     ))))


     +++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

      ((*EQ1 *PERFECT1 Y) (*EQ1 *MARY1 X))

     +++MODIFIERS+++


     PREDICATIVES->
      ((NM (ADJ ((0 0) (*PERFECT1)))) Y)

          +++ THE SEMANTIC NET +++
       *ATOM*      *VALUE*      *PROPERTY*

     PROP0106      *SOMETHING1     PRED
     PROP0106      INST0107      ARG
     PROP0105      INST0107      Y
     PROP0105      *MARY1      X
     PROP0105      *EQ1      PRED
     PRED0108      *PERFECT1      FUNC
```

```
PRED0108      INST0107      X
PROP0105      PRED0108      PRED

REMOVED PROPERTY: Y   FROM  PROP0105


+++ THE CLASSIFIED UTTERANCE IS +++

((N (NS ((0 0) (/*JOHN1)))) (A (IREG PRES ((0 0) (/*EQ1)
))) (DET ((NS INDEF)) (/*AN)) (NM (ADJ ((0 0) (/*IDEAL1)
))) (N (NS ((0 0) (/*MAN1)))))


+++ THE CLASSIFIED UTTERANCE IS +++

((N (NS ((0 0) (/*JOHN1)))) (DET ((NS INDEF)) (/*AN)
) (NM (ADJ ((0 0) (/*IDEAL1)))) (N (NS ((0 0) (/*MAN1)
))))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*EQ1 *MAN1 Y) (*EQ1 *JOHN1 X))

+++MODIFIERS+++

 ((NM (ADJ ((0 0) (*IDEAL1)))) Y)

PREDICATIVES->
 NIL
ADJECTIVES->
EXPLICIT-ADJECTIVES->
IMPLICIT-ADJECTIVES->
OPERATORS->
 (((NM (ADJ ((0 0) (*IDEAL1)))) Y))

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

PROP0110      *MAN1       PRED
PROP0110      INST0111      ARG
PROP0109      INST0111      Y
PROP0109      *JOHN1      X
PROP0109      *EQ1       PRED
PRED0112      *IDEAL1      FUNC
PRED0112      INST0111      X
PROP0109      PRED0112      PRED

REMOVED PROPERTY: Y   FROM  PROP0109
```

```
+++ THE CLASSIFIED UTTERANCE IS +++

((NM (ADJ CLASF ((0 0) (/*BIG1)))) (N (NS ((0 0) (/*MIKE1)
))) (A (IREG PRES ((0 0) (/*EQ1)))) (DET ((NS INDEF)
) (/*A)) (NM (ADJ ((0 0) (/*PERFECT1)))) (NM (ADJ CLASF
((0 0) (/*FAT1)))) (N (NS ((0 0) (/*MAN1)))))



+++ THE CLASSIFIED UTTERANCE IS +++

((NM (ADJ CLASF ((0 0) (/*BIG1)))) (N (NS ((0 0) (/*MIKE1)
))) (DET ((NS INDEF)) (/*A)) (NM (ADJ ((0 0) (/*PERFECT1)
))) (NM (ADJ CLASF ((0 0) (/*FAT1)))) (N (NS ((0 0)
 (/*MAN1)))))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*EQ1 *MAN1 Y) (*EQ1 *MIKE1 X))

+++MODIFIERS+++

 ((NM (ADJ CLASF ((0 0) (*FAT1)))) Y)
 ((NM (ADJ ((0 0) (*PERFECT1)))) Y)
 ((NM (ADJ CLASF ((0 0) (*BIG1)))) X)

PREDICATIVES->
 NIL
ADJECTIVES->
EXPLICIT-ADJECTIVES->
IMPLICIT-ADJECTIVES->
 ((NM (ADJ CLASF ((0 0) (*BIG1)))) X)
 ((NM (ADJ CLASF ((0 0) (*FAT1)))) Y)
OPERATORS->
 (((NM (ADJ ((0 0) (*PERFECT1)))) Y))

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

PROP0114      *MAN1      PRED
PROP0114      INST0115      ARG
PROP0113      INST0115      Y
PROP0113      *MIKE1      X
PROP0113      *EQ1      PRED
PROP0116      *GREATER1      PRED
INST0117      *SIZE1      MEASATTR
INST0117      *TYPVAL*      FUNC
INST0117      *SOMETHING1      CONCEPT
PROP0118      *SOMETHING1      PRED
PROP0118      *MIKE1      ARG
PROP0116      INST0117      Y
```

```
    PROP0116      INST0119      X
    INST0119      *SIZE1      FUNC
    INST0119      *MIKE1      X
    PROP0113      *GREATER1      PRED
    PROP0113      INST0120      Y
    INST0120      *MAN1      CONCEPT
    INST0120      *TYPVAL*      FUNC
    INST0120      *RELWGT1      MEASATTR

    REMOVED PROPERTY: ARG   FROM  PROP0114

    PROP0114      *MIKE1      ARG
    INST0121      *MIKE1      X
    PROP0113      INST0121      X
    INST0121      *RELWGT1      FUNC
    PRED0122      *PERFECT1      FUNC
    PRED0122      INST0124      X
    INST0124      PROP0114      CONJUNCT
    PROP0114      VAR00123      X
    INST0124      PROP0113      CONJUNCT
    INST0121      VAR00123      X
    INST0124      VAR00123      LAMBDA
    INST0124      *AND1      FUNC
    PROP0125      PRED0122      PRED
    PROP0125      *MIKE1      ARG
    NIL
=>
=>                    (UNDERSTAND)
    READY

=>                 JANE IS A PERFECT BIG TALL WOMAN.    <17>
=>
=>


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((N (NS ((0 0) (/*JANE1)))) (A (IREG PRES ((0 0) (/*EQ1)
    ))) (DET ((NS INDEF)) (/*A)) (NM (ADJ ((0 0) (/*PERFECT1)
    ))) (NM (ADJ CLASF ((0 0) (/*BIG1)))) (NM (ADJ ((0
    0) (/*TALL1)))) (N (NS ((0 0) (/*WOMAN1)))))


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((N (NS ((0 0) (/*JANE1)))) (DET ((NS INDEF)) (/*A)
    ) (NM (ADJ ((0 0) (/*PERFECT1)))) (NM (ADJ CLASF ((
    0 0) (/*BIG1)))) (NM (ADJ ((0 0) (/*TALL1)))) (N (NS
    ((0 0) (/*WOMAN1)))))


    +++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++
```

```
 ((*EQ1 *WOMAN1 Y) (*EQ1 *JANE1 X))

+++MODIFIERS+++

 ((NM (ADJ ((0 0) (*TALL1)))) Y)
 ((NM (ADJ CLASF ((0 0) (*BIG1)))) Y)
 ((NM (ADJ ((0 0) (*PERFECT1)))) Y)

PREDICATIVES->
 NIL
ADJECTIVES->
EXPLICIT-ADJECTIVES->
IMPLICIT-ADJECTIVES->
 ((NM (ADJ CLASF ((0 0) (*BIG1)))) Y)
 ((NM (ADJ ((0 0) (*TALL1)))) Y)
OPERATORS->
 (((NM (ADJ ((0 0) (*PERFECT1)))) Y))

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

PROP0127      *WOMAN1      PRED
PROP0127      INST0128      ARG
PROP0126      INST0128      Y
PROP0126      *JANE1      X
PROP0126      *EQ1      PRED
PROP0126      *GREATER1      PRED
PROP0126      INST0129      Y
INST0129      *WOMAN1      CONCEPT
INST0129      *TYPVAL*      FUNC
INST0129      *SIZE1      MEASATTR

REMOVED PROPERTY: ARG   FROM  PROP0127

PROP0127      *JANE1      ARG
INST0130      *JANE1      X
PROP0126      INST0130      X
INST0130      *SIZE1      FUNC
PROP0131      *GREATER1      PRED
INST0132      *HEIGHT1      MEASATTR
INST0132      *TYPVAL*      FUNC
INST0132      *WOMAN1      CONCEPT
PROP0131      INST0132      Y
PROP0131      INST0133      X
INST0133      *HEIGHT1      FUNC
INST0133      *JANE1      X
PRED0134      *PERFECT1      FUNC
PRED0134      INST0136      X
INST0136      PROP0127      CONJUNCT
PROP0127      VAR00135      X
INST0136      PROP0126      CONJUNCT
INST0130      VAR00135      X
```

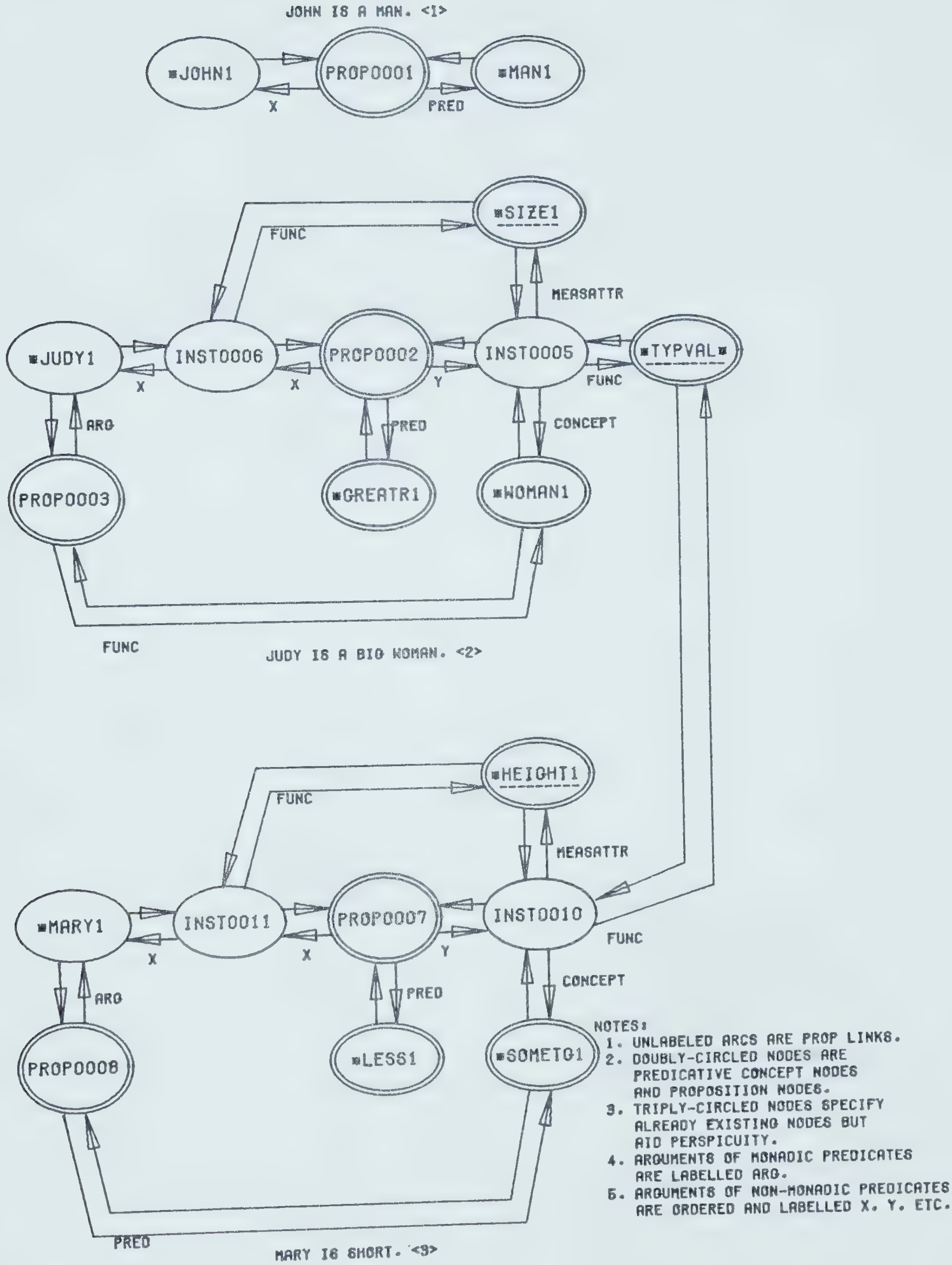
```
    INST0136     PROP0131     CONJUNCT
    INST0133     VAR00135      X
    INST0136     VAR00135     LAMBDA
    INST0136     *AND1     FUNC
    PROP0137     PRED0134      PRED
    PROP0137     *JANE1      ARG
    NIL
=>
=>                     (UNDERSTAND)
    READY

=>                     MARY IS DRINKING THE IDEAL WINE.  <18>
=>
=>


    +++ THE CLASSIFIED UTTERANCE IS +++

    ((N (NS ((0 0) (/*MARY1)))) (A (IREG PRES ((0 0) (/*EQ1)
    ))) (N (NS ((0 0) (/*DRINK6))) A (PART ((0 0) (/*DRINK1
    P1 P2)) ((0 0) (/*DRINK1A P1 P2)) ((0 0) (/*DRINK3
    P1 P2))) NM (ADJ CLASF ((0 0) (/*DRINK2 P1 P2)) ((0
     0) (/*DRINK4 P1 P2)) ((0 0) (/*DRINK5 P1 P2)) ((0
    0) (/*DRINK6 P1 P2)))) (DET ((NS NP COLL)) (/*THE))
     (NM (ADJ ((0 0) (/*IDEAL1)))) (N (NS ((0 0) (/*WINE1)
    ))))



    +++ THE CLASSIFIED UTTERANCE IS +++

    ((N (NS ((0 0) (/*MARY1)))) (N (NS ((0 0) (/*DRINK6)
    )) A (PART ((0 0) (/*DRINK1 P1 P2)) ((0 0) (/*DRINK1A
    P1 P2)) ((0 0) (/*DRINK3 P1 P2))) NM (ADJ CLASF ((0
     0) (/*DRINK2 P1 P2)) ((0 0) (/*DRINK4 P1 P2)) ((0
    0) (/*DRINK5 P1 P2)) ((0 0) (/*DRINK6 P1 P2)))) (DET
    ((NS NP COLL)) (/*THE)) (NM (ADJ ((0 0) (/*IDEAL1))
    )) (N (NS ((0 0) (/*WINE1)))))


    +++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

     ((*DRINK1A *WINE1 Y) (*DRINK1A *MARY1 X))

    +++MODIFIERS+++

     ((NM (ADJ ((0 0) (*IDEAL1)))) Y)

    PREDICATIVES->
     NIL
    ADJECTIVES->
    EXPLICIT-ADJECTIVES->
    IMPLICIT-ADJECTIVES->
```

```
OPERATORS->
 (((NM (ADJ ((0 0) (*IDEAL1)))) Y))

      +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*      *PROPERTY*

PROP0139      *WINE1       PRED
PROP0139      INST0140       ARG
PROP0141      INST0140       ARG
PROP0141      *UNS0142       PRED
PROP0138      INST0140       Y
PROP0138      *MARY1       X
PROP0138      *DRINK1A       PRED
PRED0143      *IDEAL1      FUNC
PRED0143      INST0140       X
NIL
=>
=>                    (UNDERSTAND)
READY

=>                    ANNE GAVE JUDY THE       <19>*
=>                    RED BOOK. THEN, JUDY GAVE
=>                    THE BROWN BOOK TO MARY.          <20>*
=>
=>

      *NOTE: IN THIS EXAMPLE THE ACTION IS IN
             THE PAST TENSE. SYNTACTICALLY, THIS
             IS DETECTED; SEMANTICALLY, IT IS NOT
             INCLUDED IN THE NETWORK AS YET.


+++ THE CLASSIFIED UTTERANCE IS +++

((N (NS ((0 0) (/*ANNE1)))) (A (IREG PAST ((0 0) (/*GIVE1)
))) (N (NS ((0 0) (/*JUDY1)))) (DET ((NS NP COLL))
(/*THE)) (NM (ADJ CLASF ((0 0) (/*RED1)))) (N (NS (
(0 0) (/*BOOK1)))))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*GIVE1 *BOOK1 Z) (*GIVE1 *JUDY1 Y) (*GIVE1 *ANNE1
X))

+++MODIFIERS+++

 ((NM (ADJ CLASF ((0 0) (*RED1)))) Z)

PREDICATIVES->
 NIL
ADJECTIVES->
 ((NM (ADJ CLASF ((0 0) (*RED1)))) Z)
```

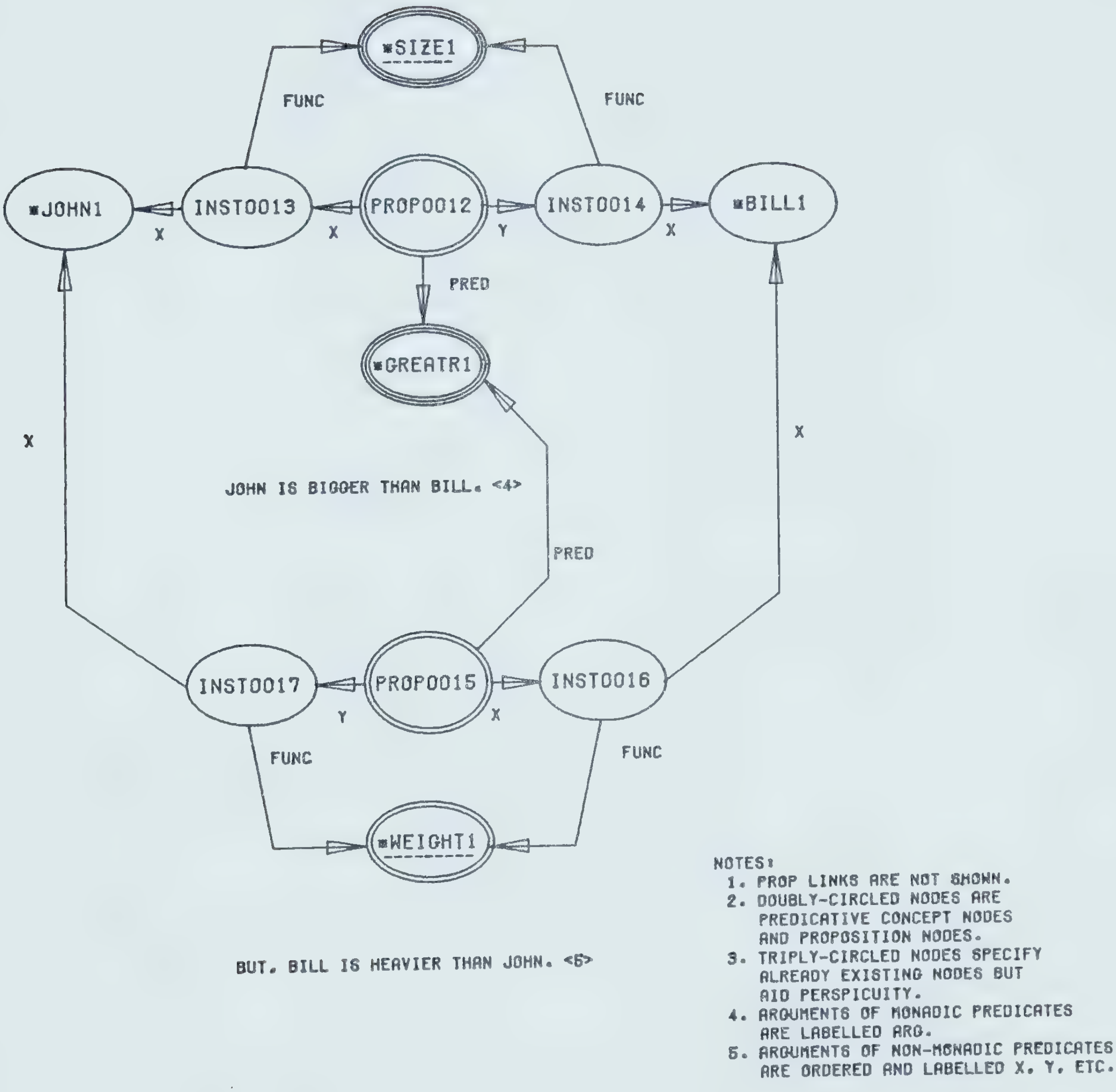
```
EXPLICIT-ADJECTIVES->
IMPLICIT-ADJECTIVES->
OPERATORS->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*     *PROPERTY*

PROP0144      *ANNE1      X
PROP0144      *JUDY1      Y
PROP0145      *BOOK1      PRED
PROP0145      INST0146      ARG
PROP0147      INST0146      ARG
PROP0147      *UNS0148      PRED
PROP0144      INST0146      Z
PROP0149      *RED1      PRED
PROP0149      INST0146      ARG
PROP0144      *GIVE1      PRED


+++ THE CLASSIFIED UTTERANCE IS +++

((AM ((ADT)) (/*THEN)) (N (NS ((0 0) (/*JUDY1)))) (
A (IREG PAST ((0 0) (/*GIVE1)))) (DET ((NS NP COLL)
) (/*THE)) (NM (ADJ CLASF ((0 0) (/*BROWN1)))) (N (
NS ((0 0) (/*BOOK1)))) (PREP NIL (/*TO1)) (N (NS ((
0 0) (/*MARY1)))))


+++ ASSOCIATED <ACTION-ARGUMENT-VARIABLE> TRIPLES +++

 ((*GIVE1 *MARY1 Y) (*GIVE1 *BOOK1 Z) (*GIVE1 *JUDY1
X))

+++MODIFIERS+++

 ((NM (ADJ CLASF ((0 0) (*BROWN1)))) Z)

PREDICATIVES->
 NIL
ADJECTIVES->
 ((NM (ADJ CLASF ((0 0) (*BROWN1)))) Z)
EXPLICIT-ADJECTIVES->
IMPLICIT-ADJECTIVES->
OPERATORS->
 NIL

     +++ THE SEMANTIC NET +++
  *ATOM*      *VALUE*     *PROPERTY*

PROP0150      *JUDY1      X
PROP0151      *BOOK1      PRED
PROP0151      INST0152      ARG
```

```
    PROPC153     INST0152      ARG
    PROP0153     *UNS0154      PRED
    PROP0150     INST0152      Z
    PROP0150     *MARY1      Y
    PROP0155     *BRCWN1      PRED
    PROP0155     INST0152      ARG
    PROP0150     *GIVE1      PRED
    NIL
=>
=>                      (MTS)
```

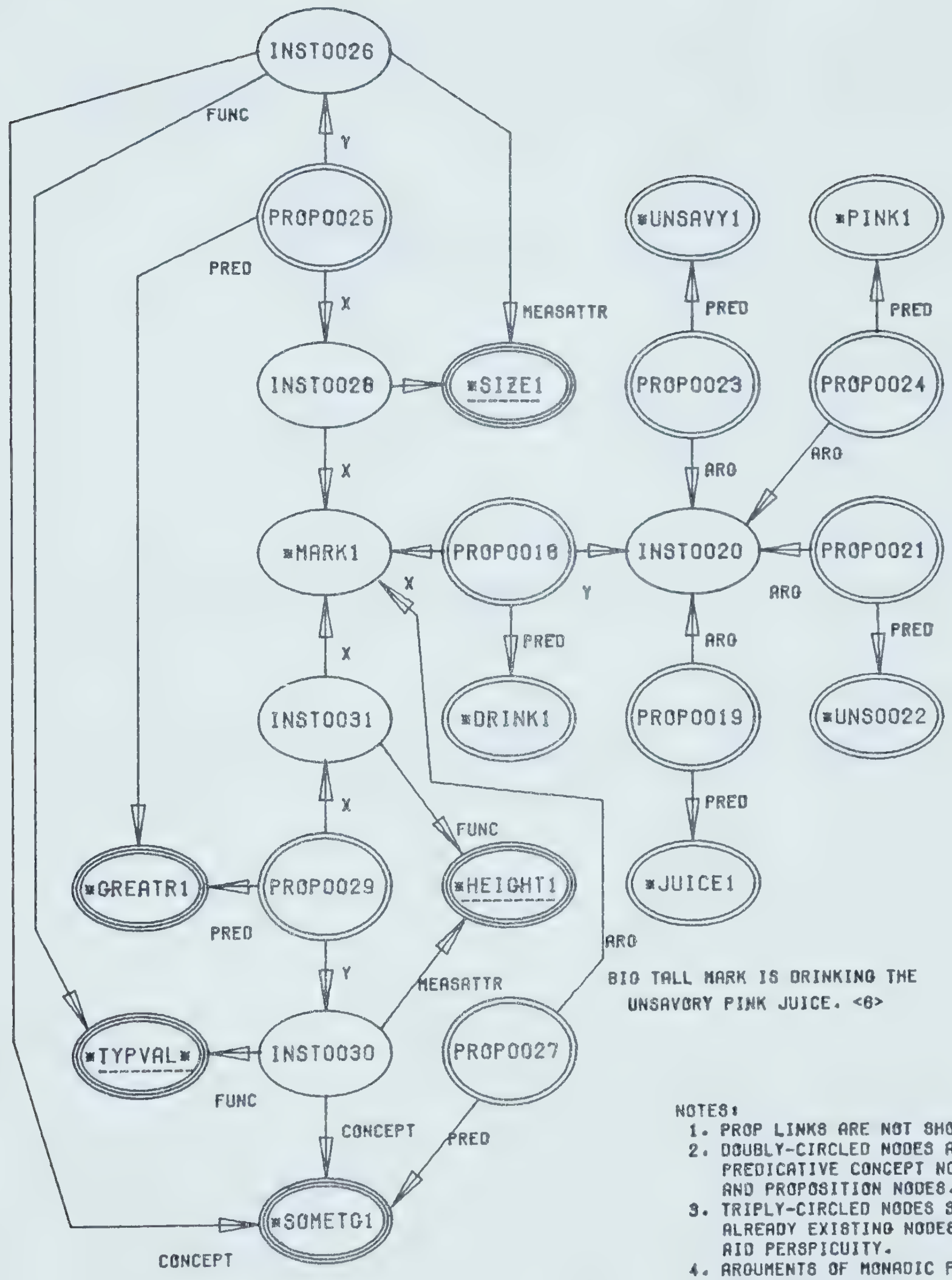
The following figures correspond to the example sentences above and are pictorial representations of the semantic networks generated for those sentences in state based theory.



D.1. Predication Networks



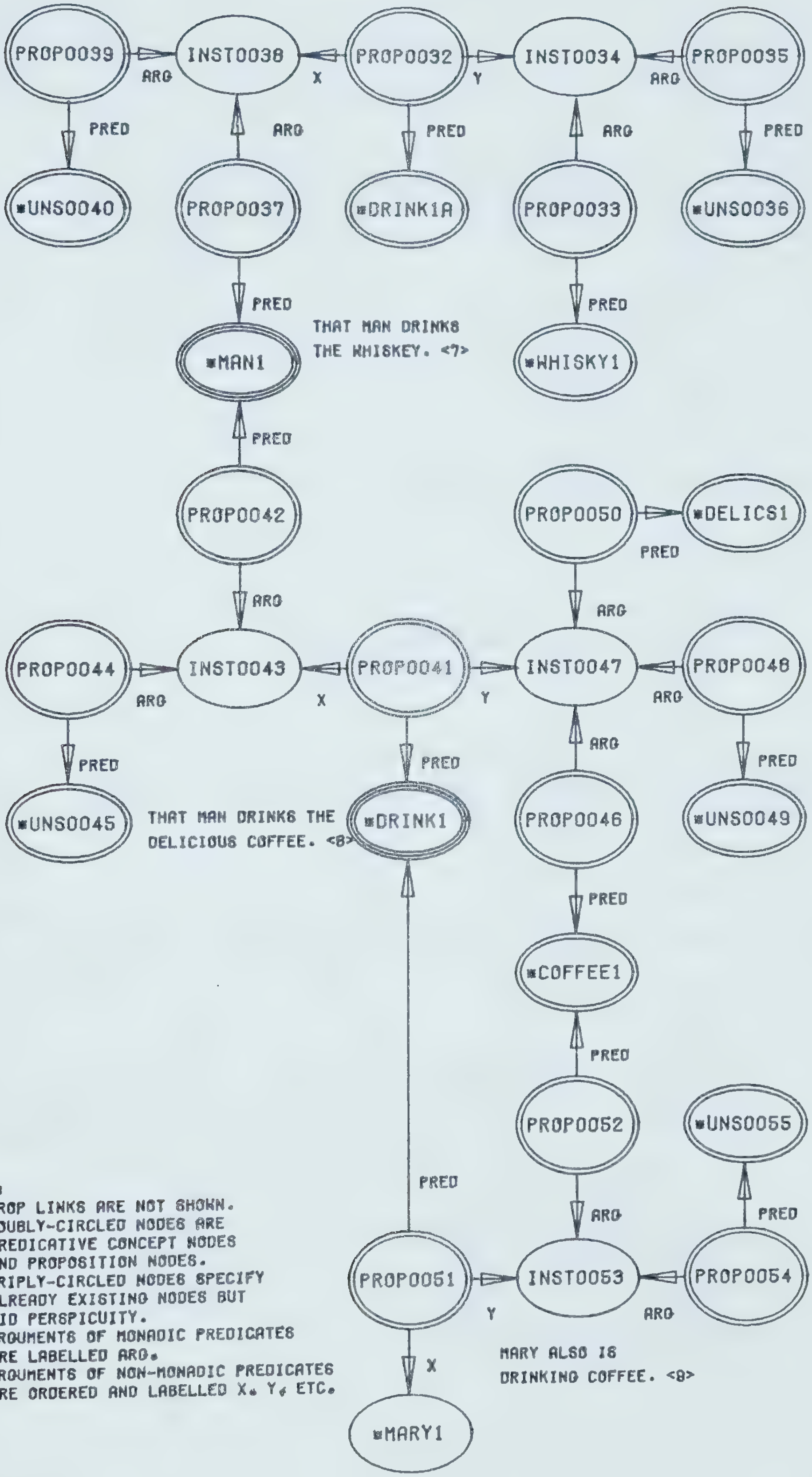
D.2. Explicit Comparative Networks



BIG TALL MARK IS DRINKING THE
UNSAVORY PINK JUICE. <6>

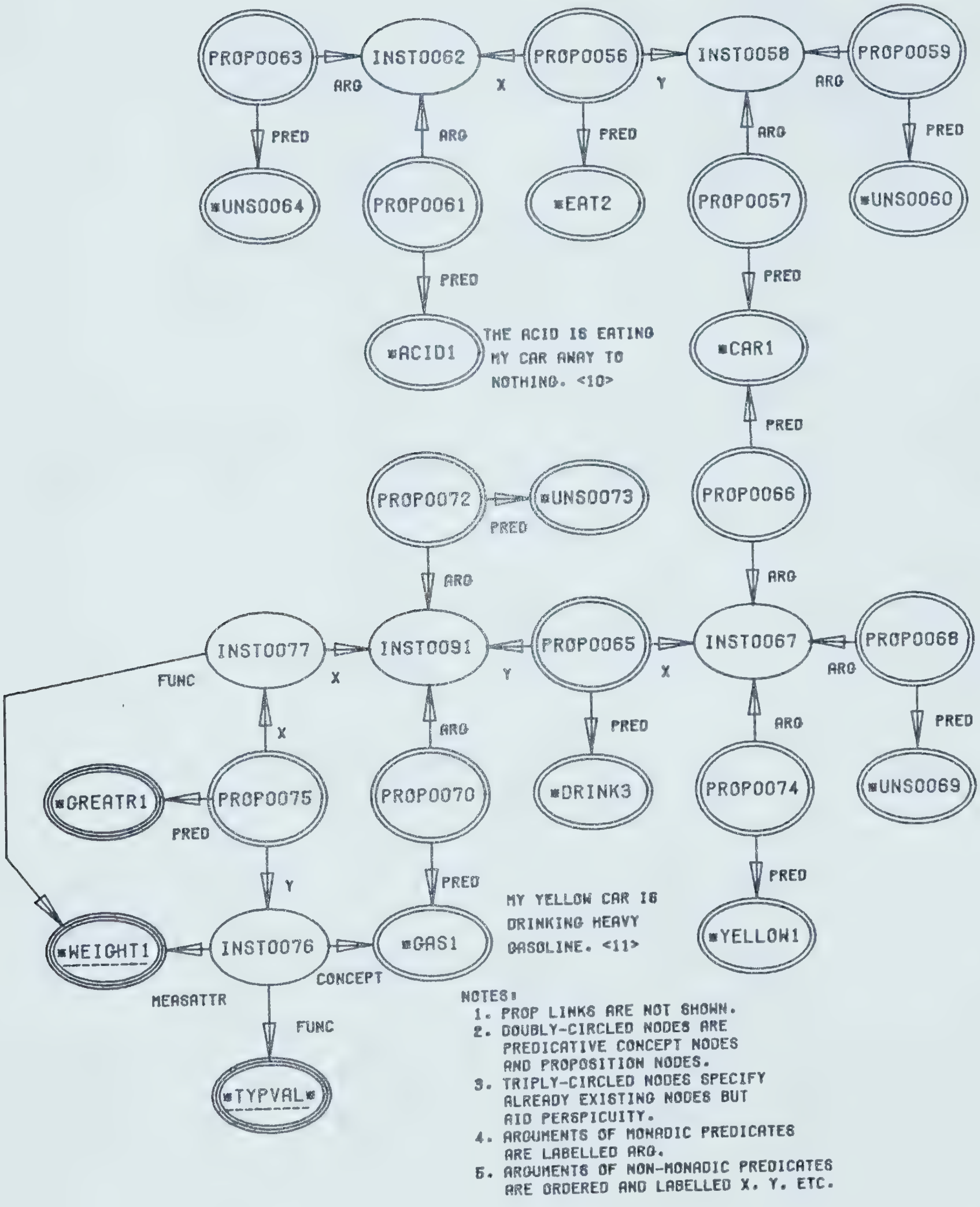
NOTES:
1. PROP LINKS ARE NOT SHOWN.
2. DOUBLY-CIRCLED NODES ARE PREDICATIVE CONCEPT NODES AND PROPOSITION NODES.
3. TRIPLY-CIRCLED NODES SPECIFY ALREADY EXISTING NODES BUT AID PERSPICUITY.
4. ARGUMENTS OF MONADIC PREDICATES ARE LABELLED ARG.
5. ARGUMENTS OF NON-MONADIC PREDICATES ARE ORDERED AND LABELLED X, Y, ETC.

D.3. Implicit Comparative Networks

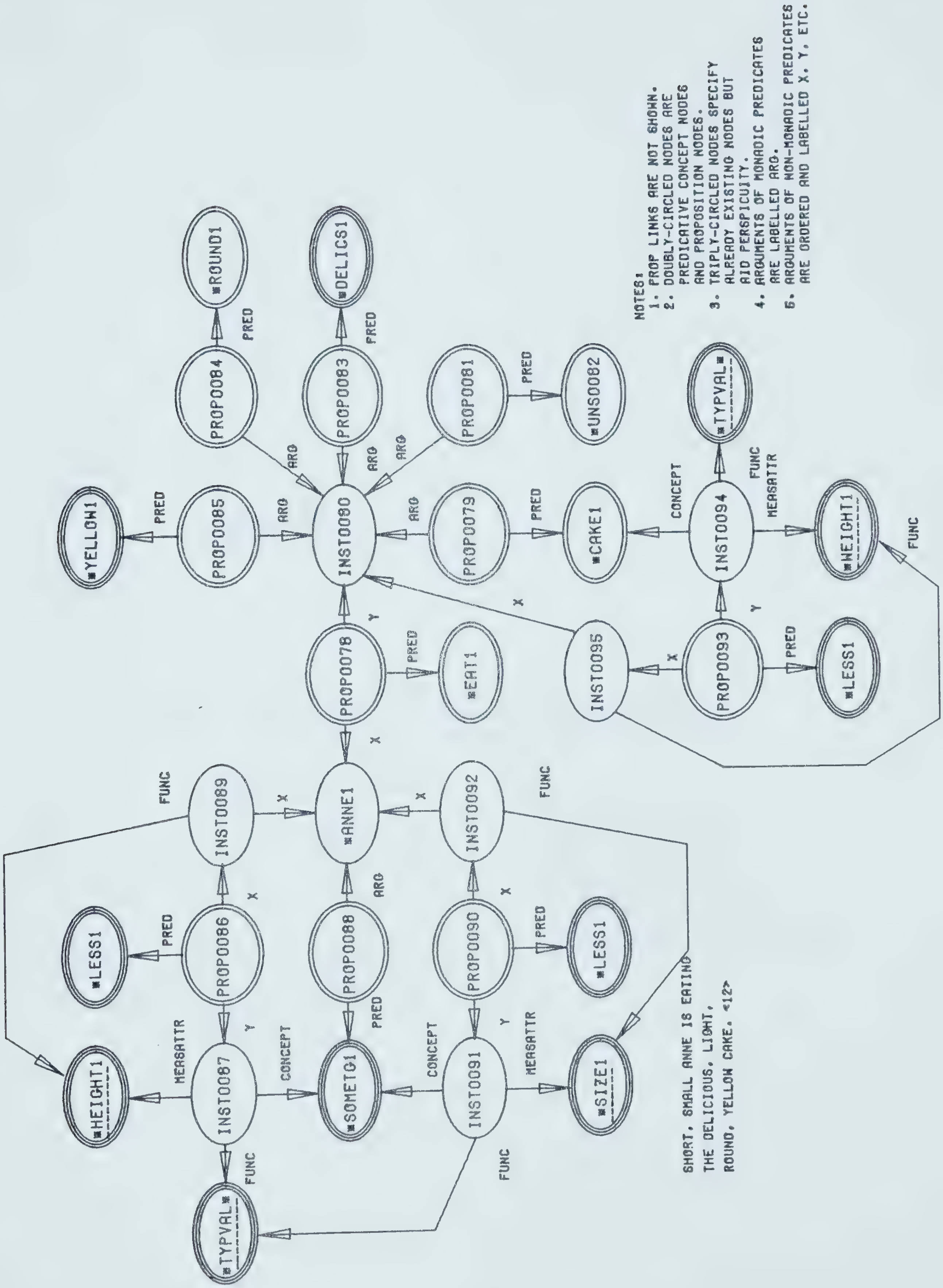


NOTES:
1. PROP LINKS ARE NOT SHOWN.
2. DOUBLY-CIRCLED NODES ARE PREDICATIVE CONCEPT NODES AND PROPOSITION NODES.
3. TRIPLY-CIRCLED NODES SPECIFY ALREADY EXISTING NODES BUT AID PERSPICUITY.
4. ARGUMENTS OF MONADIC PREDICATES ARE LABELLED ARG.
5. ARGUMENTS OF NON-MONADIC PREDICATES ARE ORDERED AND LABELLED X, Y, ETC.

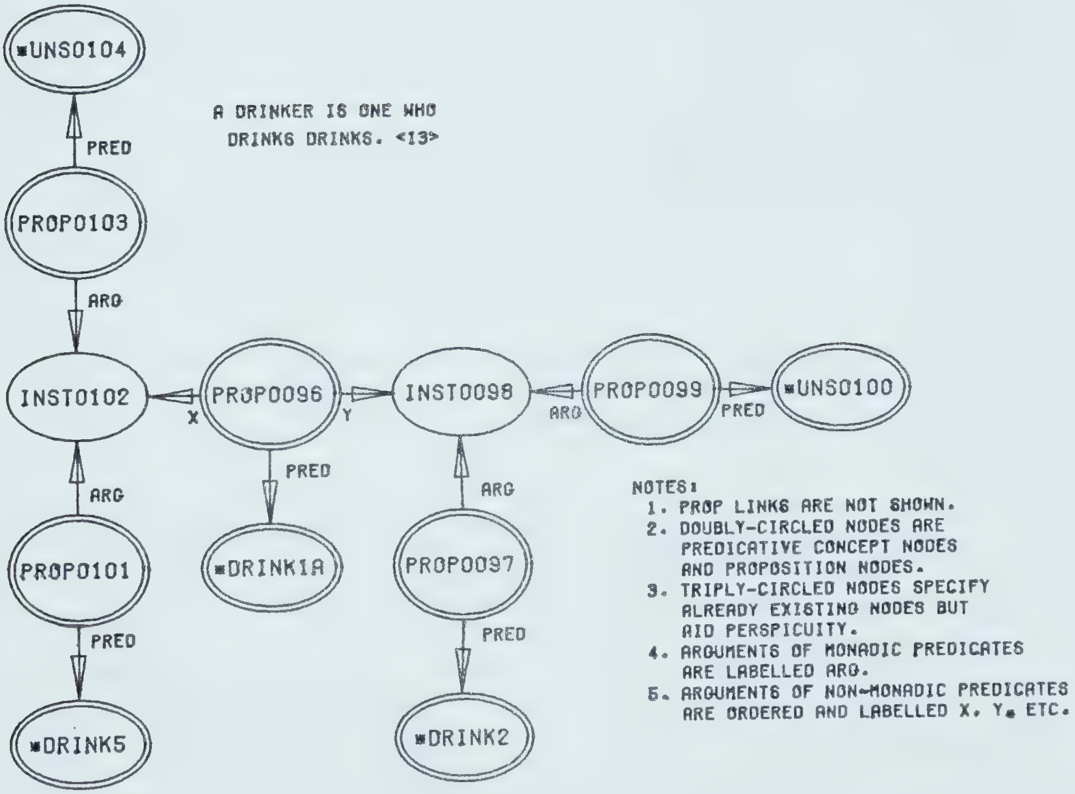
D.3. Continued



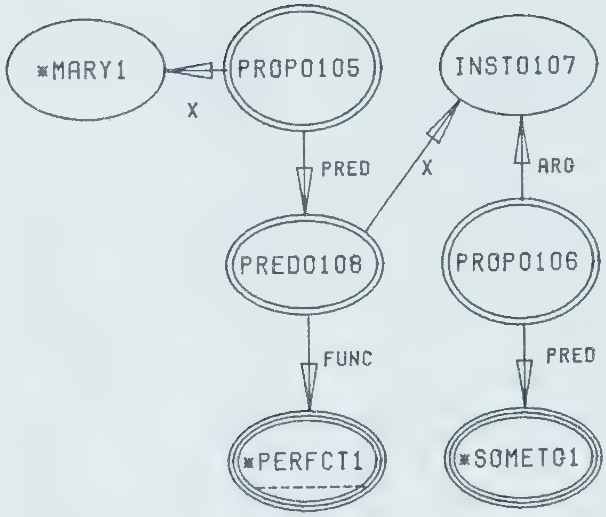
D.4. Slightly Metaphoric Networks



D.5. Mixed Bag Of Adjectives Network

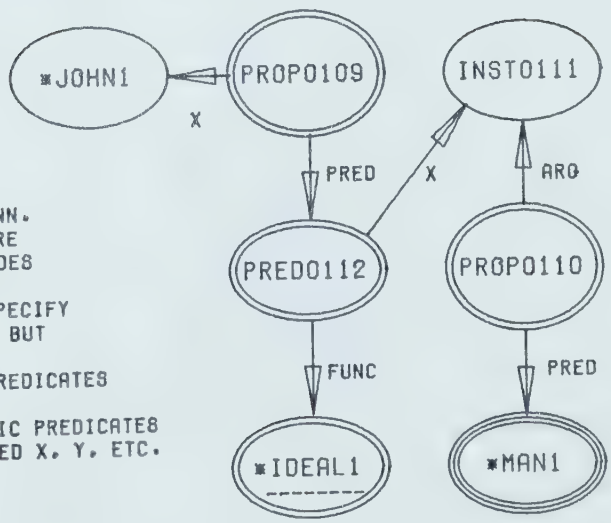


D.6. Correct Sense Network



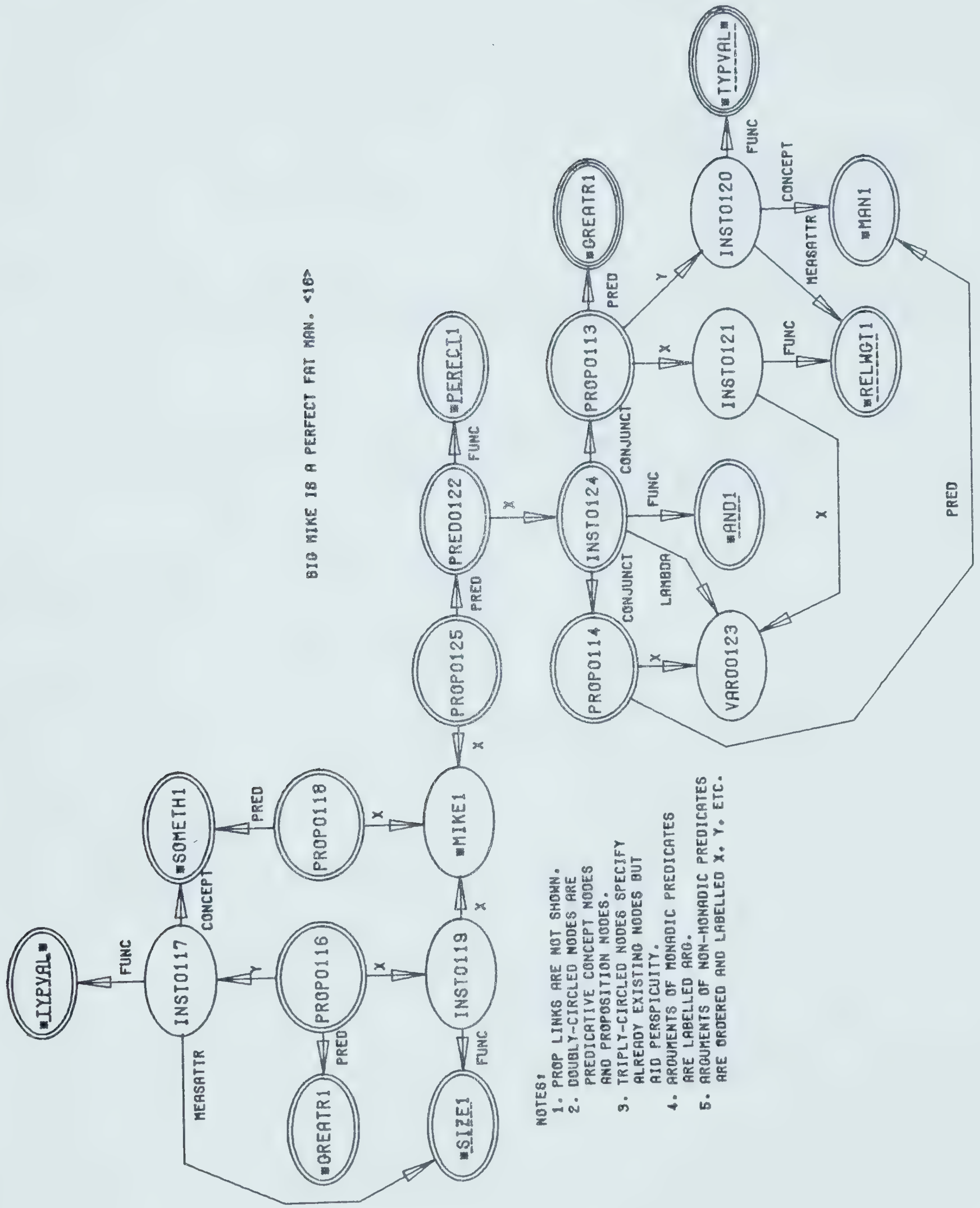
MARY IS PERFECT. <14>

JOHN IS AN IDEAL MAN. <15>

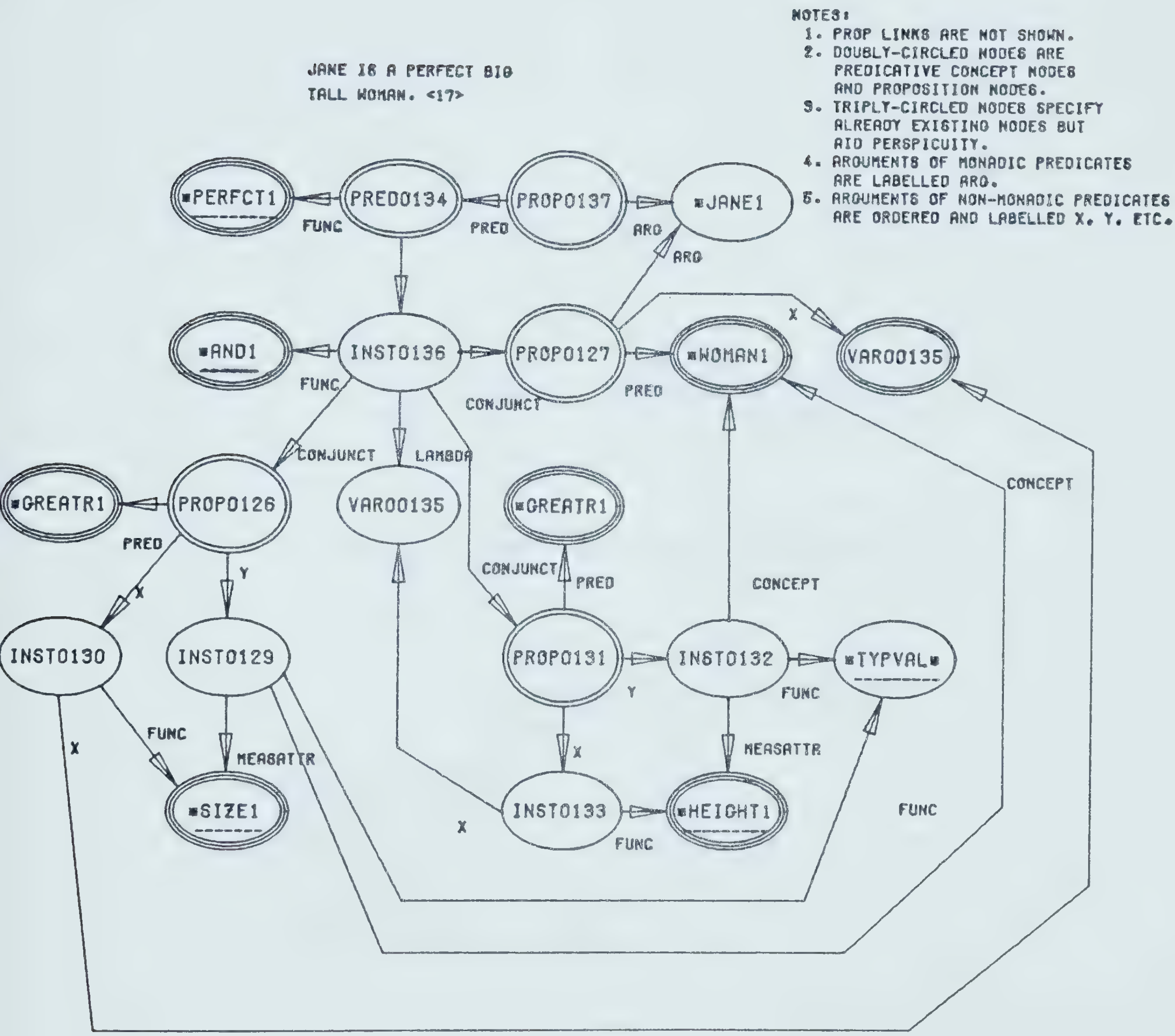


NOTES:
1. PROP LINKS ARE NOT SHOWN.
2. DOUBLY-CIRCLED NODES ARE PREDICATIVE CONCEPT NODES AND PROPOSITION NODES.
3. TRIPLY-CIRCLED NODES SPECIFY ALREADY EXISTING NODES BUT AID PERSPICUITY.
4. ARGUMENTS OF MONADIC PREDICATES ARE LABELLED ARG.
5. ARGUMENTS OF NON-MONADIC PREDICATES ARE ORDERED AND LABELLED X, Y, ETC.

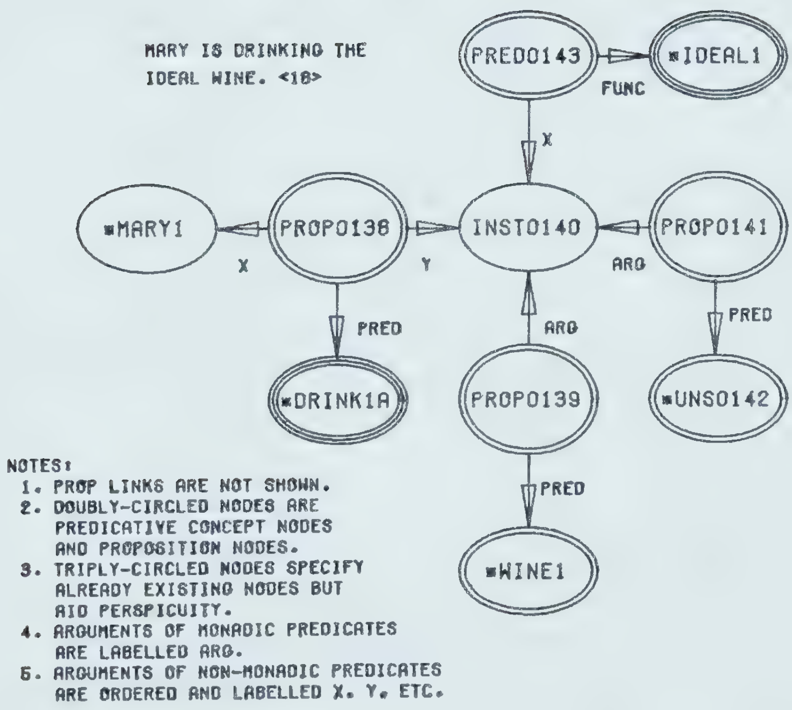
D.7. Functor Networks



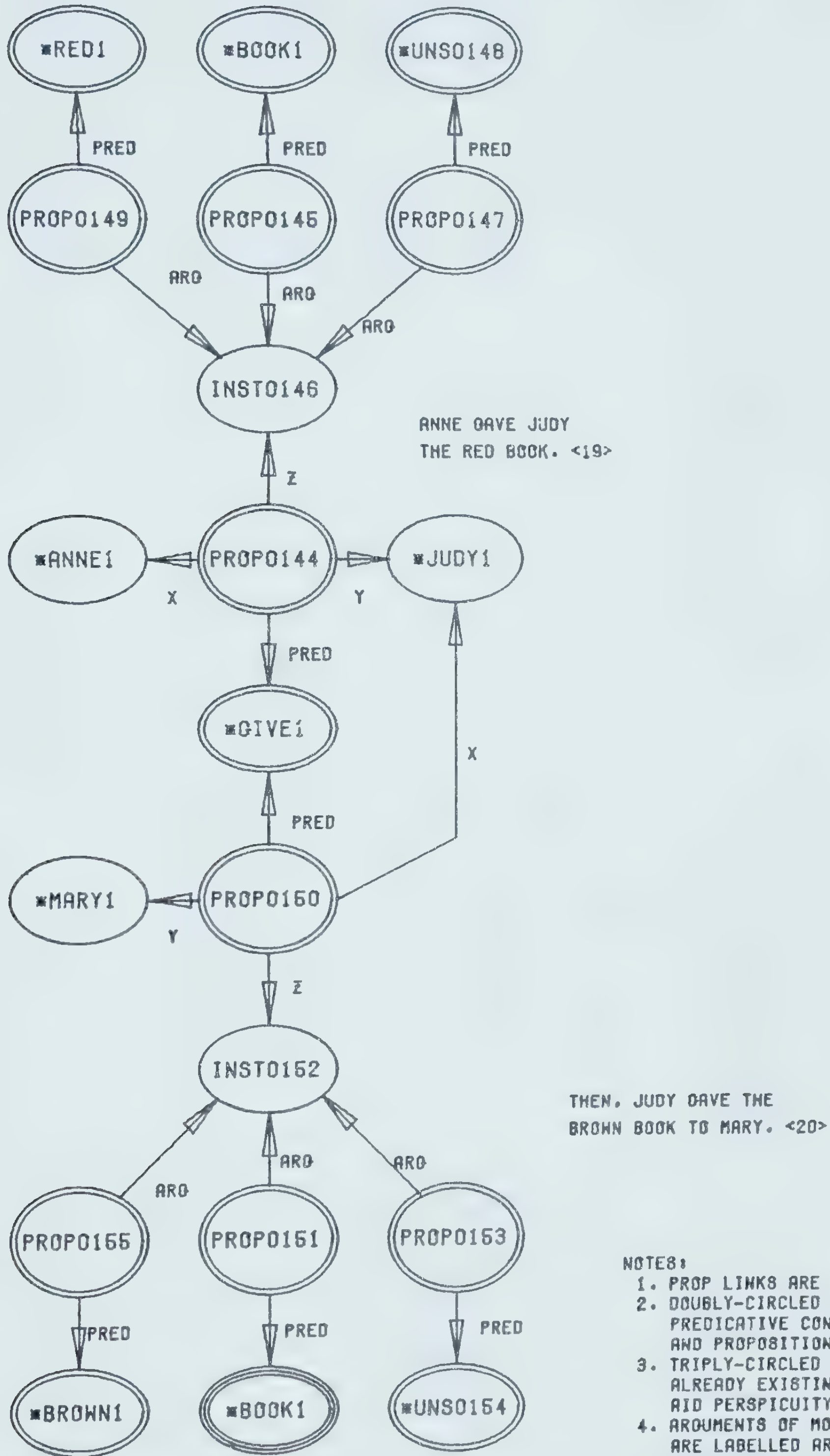
D.7. Continued



D.8. Functor And Implicit Comparative Network



D.9. Another Functor Network



NOTES:
1. PROP LINKS ARE NOT SHOWN.
2. DOUBLY-CIRCLED NODES ARE PREDICATIVE CONCEPT NODES AND PROPOSITION NODES.
3. TRIPLY-CIRCLED NODES SPECIFY ALREADY EXISTING NODES BUT AID PERSPICUITY.
4. ARGUMENTS OF MONADIC PREDICATES ARE LABELLED ARG.
5. ARGUMENTS OF NON-MONADIC PREDICATES ARE ORDERED AND LABELLED X, Y, ETC.

D.10. N-Ary Predicate Network

Appendix E System Organisation

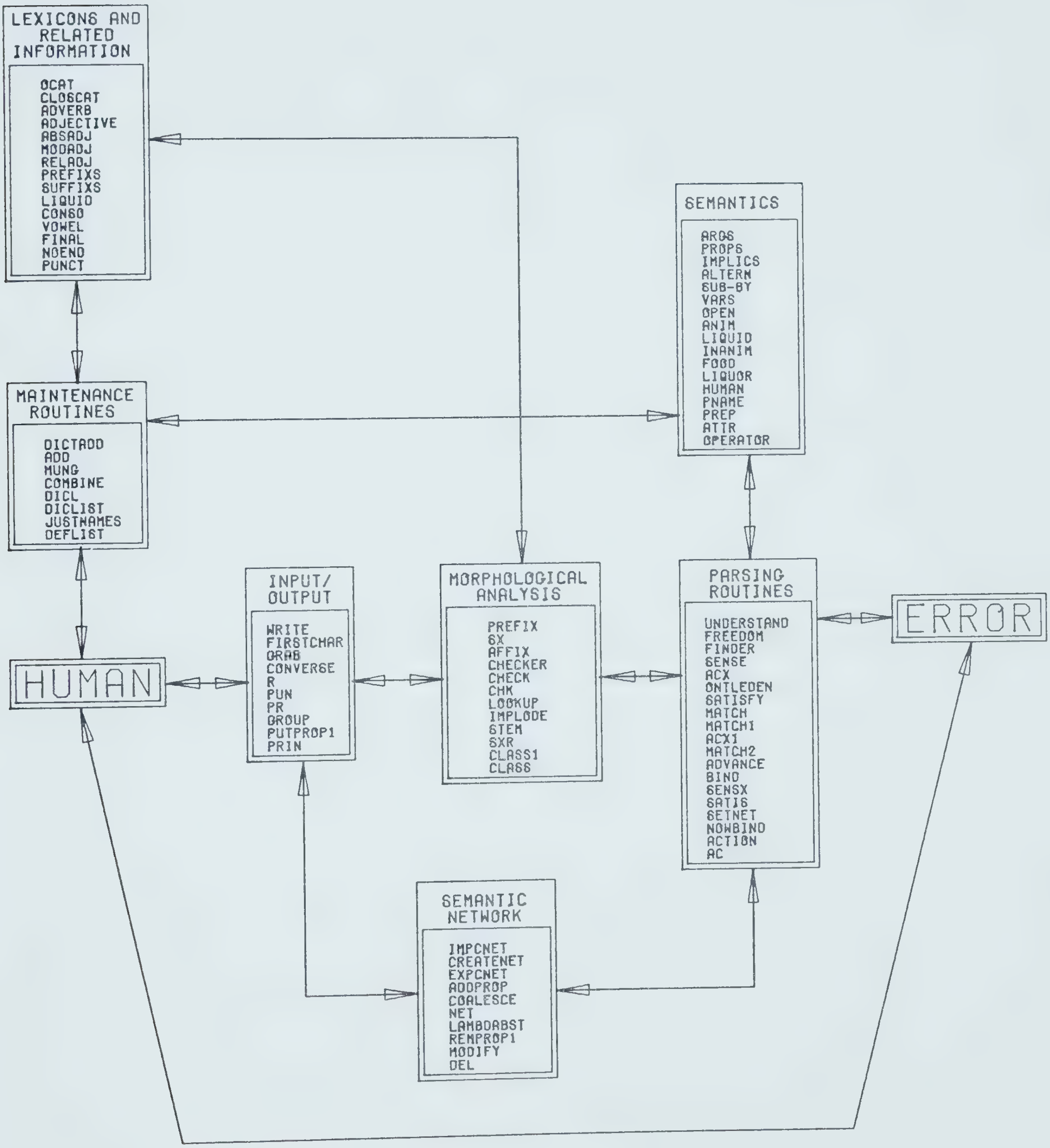


Fig. E1. A Simplified View Of The Experimental Program